Stable and Efficient Sparse Recovery for Machine Learning and Wireless Communication

Stable and Efficient Sparse Recovery for Machine Learning and Wireless Communication

A dissertation presented

by

Tsung-Han Lin

to

The School of Engineering and Applied Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Computer Science

Harvard University

Cambridge, Massachusetts

April 2014

Dissertation Advisor: Professor H. T. Kung Tsung-Han Lin

Stable and Efficient Sparse Recovery for
Machine Learning and Wireless Communication

Abstract

Recent theoretical study shows that the sparsest solution to an underdetermined linear system is unique, provided the solution vector is sufficiently sparse, and the operator matrix has sufficiently incoherent column vectors. In addition, efficient algorithms have been discovered to find such solutions. This intriguing result opens a new door for many potential applications. In this thesis, we study the design of a class of greedy algorithms that are extremely efficient, e.g., Orthogonal Matching Pursuit (OMP). These greedy algorithms suffer from a stability issue that the greedy selection approach always make locally optimal decisions, thereby easily biasing and mistaking the solutions in particular under data noise. We propose a solution approach that in designing greedy algorithms, new constraints can be devised by leveraging application-specific insights and incorporated into the algorithms. Given that sparse recovery problems by definition are underdetermined, introducing additional constraints can significantly improve the stability of greedy algorithms, yet retain their efficiency.

We demonstrate the effectiveness of the proposed solution approach in two example applications: image classification with semi-supervised machine learning, and medium access control in multiuser MIMO networks. In image classification, we show that by introducing a nonnegativity constraint in both feature dictionary learning

and feature extraction, we are able to obtain effective feature-space representations for classification purposes, especially when labeled training samples are limited. Our solution approach outperforms the classical OMP approach in classification accuracy, and is competitive with other best-known methods, while requiring much less computation. In multiuser MIMO networks, we show that sparse recovery allows us to identify transmitting host stations and estimate channel statistics from overlapping symbol sequences. The receive antenna diversity on a base station leads to a "same-support" constraint that the received signals share a same set of source transmitters. This constraint can significantly improve the convergence speed of the recovery algorithm. Moreover, this new way of concurrent channel estimation has implications on medium access strategy in multiuser MIMO for delivering throughput scalable to the available number of antennas installed on a base station.

Contents

List of Figures

List of Tables

Citations to Previously Published Work

Portions of Chapter 2 have appeared in the following paper:

T.-H. Lin and H. T. Kung, "Computing Sparse Representations in $O(N \log N)$ Time", In *Proceedings of Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS), 2013*

Large portions of Chapters 3 have appeared in the following paper:

T.-H. Lin and H. T. Kung, "Stable and Efficient Representation Learning with Nonnegativity Constraints", In *Proceedings of International Conference on Machine Learning (ICML), 2014*

Large portions of Chapters 4 have appeared in the following two papers:

T.-H. Lin and H. T. Kung, "Concurrent Channel Access and Estimation for Scalable Multiuser MIMO Networking", *Harvard University, [online] 2012, http://nrs.harvard.edu/urn-3:HUL.InstRepos:9299797*

T.-H. Lin and H. T. Kung, "Concurrent Channel Access and Estimation for Scalable Multiuser MIMO Networking", In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2013*

Large portions of Chapters 5 have appeared in on the following paper:

T.-H. Lin and H. T. Kung, "Compressive Sensing Medium Access Control for Wireless LANs", In *Proceedings of IEEE Global Communications Conference (GLOBECOM), 2012*

Acknowledgments

I would like to thank my advisor, Professor H. T. Kung, who leads me into research, guides me through the PhD study, and teaches me how to be brave to tackle the unknowns.

I would like to thank all the members of the Kung group, especially Steve Tarsa, Kevin Chen, Chit-Kwan Lin, and Dario Vlah. This dissertation would not have been possible without long-term, deep discussions with them. Most importantly, the past six years would not have been fun without them. In addition, I would like to thank Steve Tarsa and Marcus Comiter for helping edit this dissertation.

I would like to thank all my friends in Cambridge and in Taiwan. PhD study is not easy, and their support is what had kept me staying on the path.

Finally, I would like to thank my parents and my family, who always tolerate me and support my decisions.

Dedicated to my parents.

Chapter 1

Introduction

Basic linear algebra states that an underdetermined system of linear equations permits infinitely many solutions. Recently it has been discovered that a unique solution may exist if one seeks the sparsest solution. Even more strikingly, efficient algorithms exist to find such solutions [18]. The insight was first made in the signal processing community, where researchers found that overcomplete basis such as a concatenation of sinusoids and wavelets often yields sparse signal representations with superior performance in denoising and deblurring. The applicability of this intriguing idea, that one can recover high-dimensional sparse vectors from low-dimensional observations, is far-reaching beyond signal processing tasks. For instance, compressed sensing [33] suggests that data compression can be integrated into front-end sensor designs. By directly measuring low-dimensional compressed samples, the need for data storage, power consumption, and communication all can be dramatically reduced.

In this thesis, we study significant applications that are enabled with this new perspective, and also algorithms for computing sparse representations of data vectors

which are fundamental to these applications. Our study emphasizes on building empirical systems to verify the practicality of the resulting applications, and focuses on exploiting efficient greedy sparse recovery algorithms.

The two main applications we study come from completely different domains, and the nature of the sparse recovery problem is different in each. First, we tackle unsupervised representation learning for machine learning tasks, and second, we address medium access control in wireless networking. Sparsity in representation learning comes from a data analysis standpoint: one would like to explain a given data point by only a few hidden factors. Consequently, there is no ground-truth solution to the linear system, only a best approximation with respect to some desired error metric. On the other hand, in wireless networking, the sparsity is a result of a relatively small subset of transmitting host stations. Seeking for exact recovery of sparse solutions is necessary to identify the active hosts.

Despite the difference, one major lesson we learn during the course of the study across applications is the importance of *recovery stability* under noise or data variations. Substantial performance improvement in the application (e.g, higher classification accuracy or shorter computation time) can often be obtained by improving the stability of recovery algorithms. Generally, estimating high-dimensional sparse solutions in an underdetermined system is a computational problem sensitive to errors in data. This issue can be more prominent under simple greedy iterative algorithms, which gains their efficiency by taking the optimal solution via local search, and can be easily biased by noise. In statistical estimation, the stability problem is usually addressed by incorporating the noise statistics (e.g., the covariance) into the estima-

tor to prevent over-fitting. However, such noise information may be difficult to obtain in practice.

In this work, we instead make use of application-specific insights to improve recovery stability, a simple but effective alternative to statistical estimation. In particular, this knowledge is incorporated into the design of the recovery algorithm in a way that preserves its greedy algorithmic structure and its efficiency. As a result, our systems not only can demonstrate an improved performance in classification accuracy or in network data throughput, but also lead to an efficient and scalable solution. This efficiency is critical for both large-scale machine learning and communication systems.

1.1 Sparse Recovery and Applications

Consider a matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$ and $m < n$. Sparse recovery finds the solution for the following problem.

$$\min_{\mathbf{z}} \|\mathbf{z}\|_0 \quad \text{subject to} \quad \mathbf{x} = \mathbf{D}\mathbf{z} \tag{1.1}$$

Generally the solution of (2.6) is not unique, and it may seem that finding any solution will amount to an NP-hard combinatorial search. In recent years researchers have shown, however, that the solution in fact can be unique when the number of nonzero entries in $\mathbf{z}$ is sufficiently small, and the columns in $\mathbf{D}$ are sufficiently incoherent.[1]. Further, solving (2.6) is much easier than what it may seem. The solution of (2.6) can be obtained by solving an alternate problem (1.2) which turns out to have the

[1]Coherence measures the maximal correlation between all pairs of the columns in $\mathbf{D}$ (see Section 2.1)

same solution as (2.6).

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \text{subject to} \quad \mathbf{x} = \mathbf{D}\mathbf{z} \tag{1.2}$$

Equation (1.2) is a convex optimization problem and can be solved by standard linear programming [29, 27, 20]. Another approach to tackle (2.6) is to use greedy algorithms that find exactly the k nonzero entries in $\mathbf{z}$. Surprisingly, there exist simple greedy algorithms that can also obtain the solution to (2.6) [78, 79].

This recent ability to solve sparse recovery problems opens a new door for many applications. Broadly speaking, sparse recovery can be applied in two ways, one in data analysis and the other in data compression. When a given $\mathbf{x}$ is the input data of interest (e.g., image pixel intensities) and suppose a dictionary[2] $\mathbf{D}$ is available, solving (2.6) yields a *sparse decomposition* which uncovers the underlying atoms that describe the observed data. Sparse decomposition has several advantages over canonical orthogonal decomposition such as Principal Component Analysis (PCA). First, by employing an overcomplete dictionary, the decomposition can afford a larger set of candidate atoms that encode local and meaningful patterns subject to physical interpretation. Second, the sparseness constraint leads to coefficient vectors more robust to noise than the dense ones found by PCA. A dense coefficient vector in general has entries highly dependent on each other, meaning that small changes in data would alter the values of every coefficient. In contrast, when the coefficient vector is sparse, the locations of the nonzeros are likely to be kept the same. Sparse decomposition is thus particularly useful if one seeks to discover and make use of meaningful atoms, for example, for signal processing tasks such as denoising and inpainting, or analytical

[2]In linear algebra $\mathbf{D}$ is also called the "basis".

tasks such as data classification.

When applying sparse recovery for data compression, typically $\mathbf{z}$ is the sparse signal to be compressed. A shorter sketch $\mathbf{x}$ can be obtained through linear projections with projection matrix $\mathbf{D}$, which is often called the "measurement matrix". This approach is especially useful when the locations of nonzeros in $\mathbf{z}$ are not directly available, e.g., when signals are sparse in a transform domain. In this case, by using a random measurement matrix of Gaussian entries, one can compress without first transforming $\mathbf{z}$ to its sparse domain. In the context of compressed sensing, this means that one can build sensing devices to approximate the linear projection and directly measure compressed samples $\mathbf{x}$ of a shorter length m, and later reconstruct the original length-n signal of interest $\mathbf{z}$.

The two applications we study respectively fall into the two categories we discussed above. Now we briefly introduce the two applications in machine learning and wireless networking.

1.1.1 Sparse Representation Learning

Machine learning applications such as image classification requires finding a mapping between sensory data (e.g., pixel intensities) and data semantics (the "label"). This mapping can be highly complex and non-linear, and a direct modeling is very difficult. To address this issue, one typical approach is to exploit a set of *features* which captures important data characteristics, and is more directly related to data labels. By transforming the raw data into a new feature space, the similarity between data points can then be expressed in terms of the similarity between their associ-

ated features. Finding meaningful and discriminative features is the focus of many researchers in this field, either by careful feature engineering or via unsupervised feature learning.

We argue that there are two critical factors that can make this approach more powerful. One is the scale of the number of features, and the other is the sparsity in new data representations. Of the two, sparse representations can be especially beneficial. As described earlier in the section, sparse decomposition is advantageous over structural decomposition like PCA. In addition, sparse decomposition seeks to explain a given data point by only a few hidden factors. The resulting sparse representations are more linearly separable in the feature space, which simplifies supervised classifier training and makes it more likely to succeed.

This mapping from low-dimensional raw data to high-dimensional sparse representations is exactly the sparse recovery problem we discussed above. Indeed, many researchers have proposed algorithms to promote sparsity in representations and have reported higher classification accuracy in image classification. Finding sparse representations in this case, however, can still benefit from stronger theoretical justifications. As stated above, prior theoretical studies suggest that spare representations can be uniquely recovered when the overcomplete basis (the feature dictionary in this case) has sufficiently uncorrelated basis vectors and when the representation is sufficiently sparse. These two conditions in general can hardly be satisfied in practice, since a learned overcomplete dictionaries can contain similar atoms in order to improve the accuracy of data representations. Therefore, although empirically some sparsity-seeking approaches have been shown to be successful, not all of these algo-

rithms can deliver high classification accuracy consistently [22] and the causes remain unclear. The best performing approaches in terms of achieved accuracy are often convex optimization based. These approaches unfortunately are computationally expensive. Simple and efficient methods such as the greedy ones, on the other hand, are shown to achieve inferior accuracy by a significant margin.

In this thesis, we will provide some answers to this unsatisfying phenomenon. We will see that the cause of this suboptimal performance is in part due to a weak recovery stability of the greedy algorithms. By improving recovery stability, we will demonstrate competitive learning systems, which sometimes achieve even better classification accuracy on benchmark datasets.

1.1.2 Medium Access Control in MIMO Networks

In the second part of the thesis, we will switch context and study medium access control (MAC) in wireless networks. The classical medium access control problem addresses the coordination of packet transmissions from multiple host stations over a shared medium. Typically, if two packet transmissions are initiated concurrently, both packets will be dropped due to mutual interference. A strategy that can minimize collisions yet introduce only minimum overheads in delay and bandwidth is thus in need. The MAC problem has been extensively studied in the literature, with one of the most attractive solutions being *random access*. The idea of random access is to allow host stations to access the channel based on a random backoff timer, leading to fair medium sharing in a statistical sense, when the timer is appropriately chosen for the channel capacity and number of users. In this manner, the MAC

design is completely distributed and avoids employing a potentially costly centralized scheduler. By dynamically adjusting the backoff window based on channel contention levels, the packet collision probability can be minimized. For its simplicity, random access is adopted in the widely deployed 802.11 networks.

Adopting random access in multiuser MIMO networks, however, turns out to be less straight-forward due to the additional requirement of knowing information about current channel state. In MIMO networks, throughput can be linearly increased by adding antennas on the base station. The spatial diversity of the additional antennas offers extra degree-of-freedom and allow overlapping concurrent transmissions to be decoded by the receiver. To exploit the spatial diversity, the decoder needs to know the multi-path signal distortions over the wireless channels, which is also known as the channel state information (CSI). Generally CSI is measured by exchanging a known preamble sequence between transmit and receive antennas through an interference-free channel. This estimation can be easily performed in single user MIMO since all transmit antennas are co-located on the same machine and coordination between antennas is easy. In multiuser MIMO, transmit antennas are located on geographically separated stations. It is thus unclear how to employ random access, which does not explicitly schedule the transmissions of preambles and data packets, yet estimates CSI under no interference.

To address this challenge, the approach this thesis takes is to allow the preambles to be transmitted concurrently and estimate CSI using superpositions of multiple preambles. In this way, host stations can still initiate transmissions based on their own backoff timer, as in 802.11 networks without MIMO. We will see that this prob-

lem can be formulated as a sparse recovery problem, where the exploited sparsity comes from the relatively small subset of transmitting stations, and from the sparse multipath reflections. This enables a multiuser MIMO MAC design that not only supports random access, but also delivers throughput scalable to the number of available antennas on the base station.

1.2 Overview of Contributions

The potential of sparse recovery and compressed sensing has inspired a significant amount of research ranging from theoretical study to sensor hardware design. This thesis provides first-hand experiences in building applications based on *efficient* sparse recovery. Through the study, a major lesson learned is the importance of recovery stability in sparse recovery applications. Retrospectively, this finding is not surprising:

- Underconstrained systems have infinite solutions; however, under a sparsity constraint, a unique solution is guaranteed to exist.
- Greedy searches for the unique solution are fast, but susceptible to finding local minima, especially when the system is noisy.
- Introducing additional constraints restricts the search space, and reduces the likelihood that a local minimum is greedily selected.
- When these additional constraints can be derived from the application scenario, efficient recovery is possible, with improved speed and reliability.

As we will see in this thesis, based on this simple idea, substantial performance improvements in the applications can be realized.

Sparse representation learning: Although sparse coding has been widely recognized as a superior encoder for tasks like image clasification and denoising/inpainting, its success is usually at the expense of expensive computations such as solving an ℓ_1-regularized optimization problem based on convex programming. Although efficient, greedy sparse recovery algorithms do not in general give consistent performance on classification accuracy, and are especially sensitive to data variations. That is, small differences in data can make the algorithm choose a different sets of nonzero entries, making the resulting sparse representations unreliable for classification. This thesis suggests that the recovery stability can be largely improved by imposing *nonnegativity constraints* on both feature dictionaries and sparse representations. Using nonnegative models, however, means that only additive features can be modeled, but this limitation is less a problem for image data, corroborating a well-known result that nonnegative matrix factorization is capable of discovering parts-based image features [52]. Further, we will see that nonnegativity constraints can be efficiently enforced without changing the greedy algorithmic structure of the recovery algorithm, and thus the efficiency in computing sparse representations can be preserved. This allows us to build efficient large-scale image classification systems, which can deliver competitive classification accuracy on popular benchmarks compared to other best-known algorithms, including some computationally expensive approaches based on convex programming and deep neural networks.

Medium access control in multiuser MIMO networks: We will show that formulating concurrent channel estimation as sparse recovery simplifies medium access control for multiuser MIMO networks, and consequently delivers throughput scalable to the number of antennas at the base station. In addition, we will show that this recovery problem can be solved efficiently by exploiting the antenna diversity on MIMO base stations. Receiver diversity is canonically known to be useful for boosting signal SNR via maximal ratio combining. However, maximal ratio combining requires knowing the CSI a priori, and therefore is not directly applicable in concurrent channel estimation. Instead, our results show that the multiple received copies of concurrent preambles form a "same-support" constraint for sparse recovery, as the received preambles share the same set of source senders. We will see that this constraint can lead to very fast convergence of the recovery algorithm. We build a prototype of multiuser MIMO based on software-defined radios and verify the performance gain of concurrent channel estimation in practice.

1.3 Thesis Organization

The rest of the dissertation is organized as follows: Chapter 2 will introduce the basics of sparse recovery and several well-known theoretical results. We will give a novel example to show how sparse recovery can be useful in applications. In the example, exploiting the idea of sparse recovery will enable faster computations of high-level data representations. In Chapter 3 and Chapter 4, we will present our study on the two applications introduced in this chapter, namely in machine learning and wire-

less networking, respectively. Chapter 3 discusses how we can compute stable sparse data representations for machine learning tasks by sparse approximation with non-negativity constraints. Chapter 4 discusses enabling concurrent channel estimation in wireless multiuser MIMO networking by exploiting sparse recovery techniques. In Chapter 5, we will see that the idea of sparse recovery is not limited to random access based medium access control protocols. We will show an alternative MAC design that enables efficient centralized control in wireless LANs, which can be beneficial to, e.g., quality of service. Finally, we will present a summary and conclusion in Chapter 6.

Chapter 2

Sparse Recovery Basics

In this chapter, we will introduce the sparse recovery problem, provide several important theoretical results on the uniqueness guarantee of the solution, and describe popular methods for solving the recovery problem. We will keep the discussion at the high-level, with the goal of helping the readers build basic intuitions. Given that the information provided here is well established in the field, we borrow the organization and summarize from a very well-written review article by Bruckstein, Donoho, and Elad [18]. Interested readers can find more detailed information in the article. In addition, in recent years there is an explosion of research on various recovery algorithms, and this chapter is by no means thorough. We point the interested readers to [33] for a complete treatment of the subject. Finally, at the end of the chapter, we will present a simple example of applying sparse recovery for faster computation of sparse data representations that highlights the importance of sparse recovery.

2.1 Uniqueness of the Sparsest Solution

Consider a matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$ and $m < n$. Sparse recovery finds the solution for the following problem.

$$\min_{\mathbf{z}} \|\mathbf{z}\|_0 \quad \text{subject to} \quad \mathbf{x} = \mathbf{D}\mathbf{z} \tag{2.1}$$

As stated in Section 1.1, $\mathbf{x} = \mathbf{D}\mathbf{z}$ is an underdetermined system of linear equations that permits infinitely many solutions. However, the sparsest solution $\mathbf{z}$ may be unique. The uniqueness property of the solution to (2.1) can be understood by the *spark* of $\mathbf{D}$, defined as follows.

Definition 1 ([27])**.** *The spark of a given matrix* $\mathbf{D}$ *is the smallest number of columns from* $\mathbf{D}$ *that are linearly dependent.*

To see how the spark of $\mathbf{D}$ relates to the uniqueness of the solution, let us first consider a k-sparse solution vector $\mathbf{z}$ that satisfies $\mathbf{x} = \mathbf{D}\mathbf{z}$. Suppose that (2.1) permits more than one k-sparse solution, this means that we can find another k-sparse vector $\mathbf{z}'$ distinct from $\mathbf{z}$ that also satisfies $\mathbf{x} = \mathbf{D}\mathbf{z}'$. Given that $\mathbf{x} = \mathbf{D}\mathbf{z} = \mathbf{D}\mathbf{z}'$, we know that $\mathbf{z} - \mathbf{z}'$ must sit in the null space of the matrix $\mathbf{D}$, or $\mathbf{D}(\mathbf{z} - \mathbf{z}') = 0$. Now, note that both $\mathbf{z}$ and $\mathbf{z}'$ have k nonzero entries, the difference vector $\mathbf{z} - \mathbf{z}'$ should have at most $2k$ nonzero entries. This suggests that its multiplication with the matrix $\mathbf{D}$ only operates on a subset of $2k$ columns from $\mathbf{D}$. Therefore, if all possible subsets of $2k$ columns from $\mathbf{D}$ are linearly independent, we know that $\mathbf{z}$ must equal $\mathbf{z}'$.

With this observation, we are now ready to state the following result for a condition on the uniqueness of sparse solutions.

Theorem 1 ([27]). *If a system of linear equations* $\mathbf{x} = \mathbf{D}\mathbf{z}$ *has a solution* $\mathbf{z}$ *obeying* $\|\mathbf{z}\|_0 < spark(\mathbf{D})/2$*, this solution is necessarily the sparsest possible.*

Theorem 1 suggests that the uniqueness condition is governed by two factors. First, the matrix $\mathbf{D}$ needs to have sufficiently linearly independent column vectors. Second, the solution vector $\mathbf{z}$ needs to be sufficiently sparse. Later we will see that the success of recovery algorithms is also determined by these two factors.

Although spark is able to characterize the uniqueness of the sparse solution to (2.1) and in general Theorem 1 is quite sharp, computing spark can be difficult, as it requires evaluating all possible subsets of columns from a matrix. Another popular way to characterize the uniqueness property is via the *mutual coherence* of a matrix, defined as follows.

Definition 2 ([27]). *The mutual coherence of a matrix* $\mathbf{D}$ *is the maximum absolute normalized inner product between all pairs of column vectors in* $\mathbf{D}$*.*

$$\mu(\mathbf{D}) = \max_{i,j=1\ldots n, i\neq j} \frac{|\langle \mathbf{d}_i, \mathbf{d}_j \rangle|}{\|\mathbf{d}_i\|_2 \|\mathbf{d}_j\|_2} \tag{2.2}$$

Finding mutual coherence only requires computing $\mathbf{D}^T\mathbf{D}$, which is relatively easy to compute compared to spark. One can show that mutual coherence can be used to construct a lower bound for spark [27]. This allows us to find an analogue of Theorem 1 for a looser but useful bound that guarantees the uniqueness of the sparsest solution.

Theorem 2 ([27]). *If a system of linear equations* $\mathbf{x} = \mathbf{D}\mathbf{z}$ *has a solution* $\mathbf{z}$ *obeying* $\|\mathbf{z}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}))}\right)$*, this solution is necessarily the sparsest possible.*

Given the simplicity in computing and evaluating the mutual coherence, the coherence is a very popular property for various analysis of sparse recovery algorithms.

2.2 Sparse Recovery Algorithms

At the first sight, it may seem that finding any solution to (2.1) will amount to an NP-hard combinatorial search. But, researchers have found that solving (2.1) is not completely hopeless. There are two major classes of algorithms that solves (2.1) exactly under certain conditions. The first class of algorithms, termed basis pursuit (BP), relaxes (2.1) by replacing the discontinuous ℓ_0-norm by a continuous ℓ_1-norm.

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \text{subject to} \quad \mathbf{x} = \mathbf{D}\mathbf{z} \tag{2.3}$$

In this case, it can be shown that solving the ℓ_1-norm minimization problem in fact can be equivalent to solving the ℓ_0-norm minimization problem. We will later state the condition for the equivalence of the two minimization problems. Given that ℓ_1-norm is a convex function, we can solve (2.3) by standard linear programming.

The second class of algorithms, termed matching pursuit (MP), tackles the original ℓ_0-norm minimization problem by greedy iterative algorithms. Instead of performing an exhaustive combinatorial search to find the optimal k-sparse solution vector, the greedy algorithms iteratively construct a k-sparse solution vector by taking locally optimal updates. Here we focus our discussion on one of the simplest and most widely-used greedy algorithms, the orthogonal matching pursuit (OMP) [65]. OMP finds a k-sparse solution $\mathbf{z}^{(k)}$ by iterating the following steps for k rounds:

1. Initialize the residual vector $\mathbf{r}^{(0)} = \mathbf{x}$. Select the atom $\mathbf{d}_i$ that has the highest correlation value in absolute with the residual, $i_K = \arg\max_i \left|\left\langle \mathbf{d}_i, \mathbf{r}^{(K-1)} \right\rangle\right|$.

2. Approximate the coefficients of the selected atoms by least squares.
   $\mathbf{z}^{(K)} = \arg\min_{\mathbf{z}} \left\| \mathbf{x} - \sum_{l=1}^{K} \mathbf{d}_{i_l} \mathbf{z}_{i_l} \right\|_2 \; s.t. \; \mathbf{z}_{i_l} \geq 0$

3. Compute the new residual $\mathbf{r}^{(K)} = \mathbf{x} - \mathbf{D}\mathbf{z}^{(K)}$.

In OMP, the solution vector begins with an empty support set. In each iteration, a single atom is added into the support set to obtain a better approximate solution with a smaller reisdual error. The atom that can maximally reduce the residual error, which is the one that has the largest correlation value in absolute with the current residual vector, is selected. With the selected atoms, the solution vector in the K-th iteration $\mathbf{z}^{(K)}$ can be computed via the standard least squares method by only activating the selected atoms in the matrix $\mathbf{D}$. With $\mathbf{z}^{(K)}$, the residual vector then can be updated, and is carried into the next iteration to find the next approximate solution $\mathbf{z}^{(K+1)}$ that has $(K+1)$ nonzero entries. Since OMP runs for k iterations, it would always find an approximate solution with no more k nonzero entries.

Both BP and OMP can find approximate solutions to (2.1). Somewhat surprisingly, under certain conditions both approximate solutions will also be the exact solution. Even more interestingly, BP and MP will both find the exact solution under the same condition. This result is stated in the following theorem.

Theorem 3 ([27, 78])**.** *For a linear system* $\mathbf{x} = \mathbf{D}\mathbf{z}$*, if a solution* $\mathbf{z}$ *exists obeying*

$$\|\mathbf{z}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) \tag{2.4}$$

both BP and OMP are guaranteed to find the solution to (2.1) exactly.

Theorem 3 answers when and how an under-determined system of linear equations with sparse solutions can be solved – as long as the solution is sufficiently sparse, relative to the coherence of the matrix, BP and MP can solve sparse recovery exactly.

Although Theorem 3 suggests the same sufficient condition for BP and OMP to find the exact solution, in practice BP and OMP exhibit different behaviors in recovery performance. In particular, BP tends to obtain more stable solutions than OMP when noise is present (see Section 2.3 for more discussion). This might suggest that one should always use BP to solve sparse recovery problems. However, BP amounts to solving a convex optimization problem and is significantly more computationally expensive. In contrast, MP in general is very efficient and more scalable to large problems. In this thesis, for efficiency reasons, we will focus on applying MP to solve sparse recovery problems formulated in different applications. We will show that by leveraging application-specific constraints, modified MP designs are able to overcome this limitation and deliver more stable solutions or converge within a shorter amount of time.

Finally, another line of research, namely the compressive sensing [26], looks at sparse recovery with the matrix $\mathbf{D}$ as a random matrix. The entries of the random matrix are drawn identically and independently from several random distributions, including Gaussian distribution and Bernoulli distribution. For this class of sparse recovery problems, researchers have been able to show a stronger recovery guarantee [20, 60]. With high probability, both BP and MP can recover the exact solution with

$$m \geq O(k \log \frac{n}{k}) \tag{2.5}$$

In the context of compressing a sparse vector $\mathbf{z}$, the random matrix $\mathbf{D}$ can be viewed as a sensing matrix that computes a low-dimensional random projection $\mathbf{x}$, with which all the information in $\mathbf{z}$ is retained for exact recovery. (2.5) provides a lower bound for the length of the low-dimensional projection $\mathbf{x}$. More interestingly, one can

show that given a random matrix $\mathbf{D}$, exact sparse recovery is still possible even if $\mathbf{z}$ is sparse in a transform domain (e.g. an image is sparse in its DCT representation). In other words, suppose that $\mathbf{z}$ is sparse through an orthogonal transformation $\mathbf{\Psi}$ that $\mathbf{z} = \mathbf{\Psi s}$, the following sparse recovery problem still can be solved with the same recovery guarantee.

$$\min_{\mathbf{s}} \|\mathbf{s}\|_0 \quad \text{subject to} \quad \mathbf{x} = \mathbf{Dz} = \mathbf{D\Psi s} \tag{2.6}$$

This suggests that this type of compression by random linear projections can be performed directly on the raw data ($\mathbf{z}$), and there is no need to first transform data to its sparse representations ($\mathbf{s}$) in the transform domain. In Chapter 4 and 5, we will employ this type of compression with random projections to facilitate medium access control in wireless networks.

2.3 Stability of Sparse Recovery Under Noise

We have seen in the previous two sections that sparse recovery problems are not completely hopeless and there exist algorithms to find the unique solution. In this section, we consider a more general setting where noise is present in the data. In practice, noise is inevitable, and a natural question is to ask, "how stable is a sparse recovery algorithm in this setting?"

When data is presented with noise, we may solve an noise-aware formulation of the recovery problem, with tolerance ϵ:

$$\min_{\mathbf{z}} \|\mathbf{z}\|_0 \quad \text{subject to} \quad \|\mathbf{x} - \mathbf{Dz}\|_2 \leq \epsilon \tag{2.7}$$

Similarly to BP, we can relax (2.7) by replacing the ℓ_0-norm by ℓ_1-norm and obtain the following problem, which is also known as basis pursuit denoising (BPDN) [21]:

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \text{subject to} \quad \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2 \leq \epsilon \tag{2.8}$$

There exists a wide range of methods including convex optimization techniques to solve BPDN. Here we will not go into details of these methods, but instead we highlight the stability result of the solution in BPDN.

Theorem 4 ([28])**.** *Suppose that a vector* $\mathbf{z}$ *satisfies the sparsity constraint* $\|\mathbf{z}\|_0 < (1 + 1/\mu(\mathbf{D}))/4$ *and* $\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2 \leq \epsilon$*. The solution* $\mathbf{z}^*$ *of (2.8) must obey*

$$\|\mathbf{z}^* - \mathbf{z}\|_2^2 \leq \frac{4\epsilon^2}{1 - \mu(\mathbf{D})(4\|\mathbf{z}\|_0 - 1)} \tag{2.9}$$

This result shows that BPDN is quite stable under noise. The error in the computed sparse vector $\mathbf{z}^*$ only grows proportionally to the noise.

On the other hand, we can solve (2.7) with OMP by introducing a stopping rule to terminate OMP earlier if needed: when the ℓ_2-norm of the residual vector is smaller than the noise level (ϵ), OMP would terminate. OMP exhibits a weaker stability that it is only "locally stable," i.e., the solution is only stable under small noise. Under large noise, OMP is doomed to fail due to the greedy atom selection procedure. In this case, large noise would cause a wrong set of k atoms to be selected and there is no hope to bound the error in the solution. The following result describes the local stability behavior of OMP. In order to have OMP select the correct set of atoms, the noise has to be sufficiently small compared to the magnitude of the smallest nonzero entry in $\mathbf{z}$.

Theorem 5 ([28]). *Suppose that a vector* $\mathbf{z}$ *satisfies* $\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2 \leq \epsilon$ *and*

$$\|\mathbf{z}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) - \frac{\epsilon}{\mu(\mathbf{D})\,|z_{min}|} \tag{2.10}$$

where z_{min} *is the smallest nonzero entry in* $\mathbf{z}$*, OMP is guaranteed to recover a solution with the correct support.*

There have been several proposals of MP algorithms that can improve OMP's stability by paying a higher computation cost. These include ROMP [61], StOMP [31], and CoSaMP [60]. Typical strategies include selecting more than one atom in each iteration, allowing atom replacement between iterations, and using a more careful atom selection scheme that avoids choosing atoms that have overly small correlations with the residual.

2.4 Application: Computing High-Level Data Representations in Compressed Domain

Finally, we present a simple application example that makes use of the sparse recovery discussed in this chapter. This application concerns finding high-level sparse data representations for given input data. With such representations, machine learning methods like SVM are more likely succeed. We will show that the computation efficiency for computing such representations can be improved by moving the computations to a *compressed domain*.

A recent insight based on *deep learning* [10] calls for stacking multiple layers of nonlinear operations into an inference hierarchy to process input signals and extract

representations. Such deep architectures have attained state-of-the-art performance in some applications, e.g., computer vision and speech recognition. In these methods, representations computed in a preceding layer are taken as input to the next layer to form higher-level representations. In image processing, this corresponds to progressively describe objects using features of larger spatial scales. For example, in the bottom layer objects can be represented by edges of different widths, lengths and orientations over small regions. In higher layers objects may be described by shapes such as squares and triangles over large regions.

We assume that the representations are computed based on overcomplete feature dictionaries, such that input signals can be represented using just a few dictionary atoms. Therefore, finding the representations is a sparse recovery problem introduced in this chapter. In order to process a large amount of data, we consider using OMP for its high efficiency to compute the sparse representations. Although OMP is already very efficient, the computational cost is proportional to the dimension of the columns vectors in the dictionary, and as we will see, this dimension grows very fast in a deep architecture. Computing high-level representation is therefore still a very time-consuming task.

Let us illustrate this problem with an example. Suppose that the input data is an $m \times 1$ vector $\mathbf{x}$, and the dictionary has n of the $m \times 1$ dictionary atoms. Then the bulk of the sparse representation computation amounts to computing n correlations between $\mathbf{x}$ and each of the n atoms, which is $O(mn)$. Note that in this cost, m and n are governed by different factors. At the first layer of the hierarchy, m is driven by input data dimension, while at subsequent layers it is driven by the complexity

of intermediate feature representations. Moreover, as feature representations grow in spatial scale, m typically increases further up the hierarchy. For example, independent patches of a 4×4 grid are transformed at Layer 1, m increases by 16× when these are vectorized in Layer 2. On the other hand, n is typically governed by the characteristics of a given machine learning task. If the task is to classify objects with a large number of categories, n tends to be large for an increased chance of representing an input $\mathbf{x}$ with just a few dictionary atoms. On the other hand, with an easier task such as differentiating between only a few simple objects, a relatively small n may be sufficient to derive discernible representations.

We show that this $O(mn)$ cost can be reduced to $O(n \log n)$, a complexity independent of the data or representation dimension m, by moving the computations to a *compressed domain*. This means that the reduced computation cost is only dictated by the desired classification resolution.

2.4.1 Performing OMP in the compressed domain

To realize this gain, we propose to perform OMP's operations in a lower dimensional subspace, i.e., a compressed domain, thereby significantly reducing the data size and computation cost. We describe our results for a data vector $\mathbf{x}$ that can be exactly represented using k atoms from a normalized $m \times n$ dictionary $\mathbf{D}$, meaning that $\mathbf{x} = \mathbf{D}\mathbf{z}$, where $\mathbf{z}$ is the sparse representation. We define a linear mapping $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}^{\widehat{m}}$ with $\widehat{m} < m$ and call the reduced-dimension problem of recovering $\mathbf{z}$ from $\Phi\mathbf{x} = \Phi\mathbf{D}\mathbf{z}$ the *compressed domain recovery problem.*

Theorem 6. *Suppose that the dictionary is sufficiently incoherent, i.e., obeying*

$$\mu(\mathbf{D}) < \frac{1}{2k-1} - \delta \quad \textit{for some} \quad 0 < \delta < 1 \tag{2.11}$$

Then OMP solves the compressed domain recovery problem, where $\widehat{m}$ can be as small as $O(\log n/\delta^2)$.

Proof. We can prove this theorem by leveraging the Johnson-Lindenstrauss lemma [47]. The Johnson-Lindenstrauss lemma states that with high probability, a random linear mapping Φ can embed the n dictionary atoms of any dimensionality to an $O(\log n/\delta^2)$-dimensional subspace in a way that the distances between any pairs of atoms are nearly preserved, subject to a small distortion factor δ. In other words, the inner product between any two atoms is also preserved by up to a distortion of δ in the subspace [25]. This means that the mutual coherence of the compressed dictionary $\mu(\Phi\mathbf{D})$ will not be changed by more than δ from $\mu(\mathbf{D})$. Theorem 6 then follows from the sufficient condition for exact recovery stated in Theorem 3. □

By Theorem 6, for constant δ and k, $\widehat{m}$ can be in the order of $O(\log n)$. Since $\widehat{m}n = O(n \log n)$, the computation cost of OMP can now be reduced to $O(n \log n)$.

2.4.2 Compressed Domain OMP in Image Classification

We now evaluate the effectiveness of compressed domain OMP in computing representations for image classification purpose. The architecture employed in our implementation to compute the representations is similar to that in Hierarchical Matching Pursuit [16], and our architecture has two layers. The sparse representations are computed over 6×6 overlapping grids in Layer 1; these representations are vectorized

in Layer 2 over 16×16 grids and sparse-coded again to form higher-level representations. The resulting Layer 2 representations are then aggregated over the whole image by max-pooling (taking component-wise maximum of all representations) to form an image representation. The dictionaries at Layer 1 and 2 are set to have dimensions 36×108 and 2268×1000, and the sparsity k in representations is set to 5 and 10, respectively. The dictionaries are built through unsupervised dictionary learning as detailed in [16].

We perform the evaluation using the Caltech-101 dataset [34] that has 101 object categories. After transforming raw pixel data into image representations, we use 30 training images in each category to train a linear SVM (L2-SVM), and use no more than 50 images in each category for testing. For comparison, we compute the Layer 2 representations using three different Layer 2 dictionaries. The first one is the orignal dictionary of size 2268×1000, and the other two are constructed by randomly projecting the original dictionary to a lower-dimensional subspace using random Bernoulli matrices. The two compressed dictionaries have size 1134×1000 and 226×1000, corresponding to 2× and 10× compression ratio.

The experiment results are summarized as follows. Without any compression, we are able to obtain 59.9% classification accuracy. With 2x and 10x compression, the classification accuracy drops slightly to 59.3% and 56.7%, respectively. This suggests that the compressed domain approach can effectively reduce the computations in computing high-layer data representations, with only a small effect on classification accuracy. For example, the computational cost of OMP can be reduced by 10× with less than 3% decrease in classification accuracy.

Chapter 3

Sparse Representation Learning

In this chapter, we will discuss using the sparsity constraints for computing feature-space data representations. In machine learning, proper data representations are often the key to the success of learning methods. Good data representations usually make use of a set of features that capture important data statistics. Such features will also tie closer to data semantics. With these feature-space representations, the similarity between two data instances can be expressed in terms of the similarity between their corresponding features. As a result, machine learning tasks such as classification can be simplified and are more likely to succeed.

Orthogonal Matching Pursuit (OMP) is a popular method for computing the feature-space data representations. The data representations computed by the classical OMP encoder, however, are highly unstable, i.e., the representations vary dramatically under small data variations or data noise, and the poor stability will hamper the effectiveness of the computed representations for classification. We will see that this stability issue can be largely alleviated by introducing the nonnegativity constraints

on both features and representations. We will propose a variant of OMP, named the nonnegative OMP (NOMP), that incorporates the nonnegativity constraints in computing sparse data representations. NOMP is an efficient encoder, scalable to large feature dictionaries. In our experiments, NOMP consistently outperforms OMP in classification accuracy by large margins on several popular benchmark datasets. Nevertheless, NOMP is competitive with other best-known feature extraction methods in classification accuracy, particularly when the available labeled training samples are limited.

3.1 Introduction

We consider computing high-level image representations with which we can more easily classify images. Such high-level representations are typically derived by encoding low-level image descriptors into a suitable feature space based on a feature dictionary. Much work has been devoted to unsupervised feature dictionary learning over the past years (see [11]). Recently, it has been shown that the K-means algorithm is usually sufficient for this task [22], providing a very efficient solution for dictionary learning.

On the contrary, efficient encoder design for computing data representations based on learned dictionaries has received less attention. A good encoder usually finds representations that are *sparse*, with the hope that the new representations are linearly separable in the new feature space and will simplify classifier training. A good encoder usually finds representations that are *sparse*, with the hope that the new representations are linearly separable in the feature space and will simplify classifier training.

Imposing this sparse prior, however, often invokes a considerable amount of computations. For example, the classical approach to sparse coding involves solving an expensive ℓ_1 minimization problem [53, 66], which is less applicable for large-scale machine learning problems.

There have been several attempts to use efficient approximation algorithms for sparse encoding (see [22]). One example is the soft-threshold encoder, which finds sparse representations by simply dropping entries smaller than a certain threshold [59, 49]. Such encoder has been shown to work well in benchmarks containing abundant labeled training samples. In contrast, efficient greedy algorithms, such as Orthogonal Matching Pursuit (OMP) [65], are less successful in computing effective representations. OMP is reported to deliver suboptimal classification accuracy on popular benchmarks.

In this work, we show that OMP in fact is *not* a poor encoder. We have found that a key to making OMP perform well is to introduce *nonnegativity constraints*. Nonnegativity constraints have long been exploited for learning sparse, additive features. For example, nonnegative matrix factorization (NMF) has been shown to learn parts-based representations [52]. By further including sparseness constraints into NMF, it has been observed that Gabor-like low-level features can be learned [44]. In addition, nonnegativity contraints are biologically plausible for modeling human vision systems in computational neuroscience research [43]. However, despite the large corpus of nonnegative feature learning algorithms in the literature, little is known about the utility of nonnegativity constraints in encoding sparse representations.

We found that imposing nonnegativity constraints can largely alleviate a stability

issue of OMP, namely that OMP may fail to find nearby representations for data with small variations [28, 69]. The instability of the computed representations can lead to confusions in classifier training and inferior classification accuracy. We argue that under nonnegativity constraints, OMP's stability is enhanced, and in addition, will increase with pairwise separation among dictionary atoms. This means that with a better trained dictionary where atoms are well separated, the encoder can be much more stable.

We have validated the effectiveness of the nonnegative OMP encoder (NOMP), a variant of OMP that is as efficient, through experiments on the CIFAR-10, CIFAR-100 [51], and STL-10 datasets [24]. We present two major findings:

- The proposed NOMP encoder outperforms the prior OMP encoder in classification accuracy by large margins. Like prior sparsity-seeking encoders such as OMP, NOMP can tackle datasets containing a small amount of labeled training data. In contrast, to achieve comparable accuracy performance, other fast feed-forward encoders, such as the soft-threshold encoder, would have to use supervised classifier training involving substantially more labeled training data.

- With a moderate amount of labeled training samples, NOMP is competitive in classification accuracy with the state-of-the-art deep neural networks, and is much faster and easier to train.

3.2 Related Work

Sparse coding is a promising method for object classification (e.g.,[67]). Coates and Ng [22] point out that the effectiveness of sparse coding is contributed largely by its encoding capability that finds sparse data representations. As discussed in Chapter 2, a common strategy to promote sparsity in representations is ℓ_1-regularization [63]. However, to encourage sparsity in representations, solving the related ℓ_1-minimization problem can be computationally expensive. A considerable amount of work is dedicated to designing efficient ℓ_1-minimization algorithms (e.g., [53]).

For computational efficiency, researchers have also developed fast nonlinear encoders, such as the *tanh* function, to compute sparse solutions. In particular, these nonlinear encoders may be trained to approximate solutions computed by ℓ_1-sparse coding [48, 41]. Moreover, it has been shown that the simple soft-threshold encoder, $\max(0, \mathbf{D}^T\mathbf{x} - \alpha)$ for some small $\alpha > 0$, can be competitive in some cases [59, 49, 22]. In this work, we take a different path in which OMP is employed to encode sparse representations.

Nonnegative matrix factorization [64, 52] and nonnegative sparse coding [44] are related feature extraction methods that enjoy much empirical success. While their use is often motivated by the nonnegative nature of applications (for example, document analysis), theoretical studies suggest that nonnegativity constraints themselves can be powerful. It has been shown that nonnegativity constraints can ensure a unique sparse solution without ℓ_1-regularization [30, 19]. Slawski and Hein [73] further show that thresholded nonnegative least squares can be resistant to overfitting of noise even in underdetermined sparse recovery.

Following this line of research, we show that nonnegativity constraints can be useful when efficient approximation algorithms such as OMP are used in computing sparse representations, especially for object classification purposes. We are not the first to propose a nonnegative variant for OMP. In fact, nonnegative extensions have been repeatedly proposed in the literature [19, 72]. However, as far as we know, we are the first to identify and analyze the stability advantage of nonnegativity constraints for OMP.

3.3 Encoding Sparse Representation with Nonnegativity Constraints

Suppose that we are given a feature dictionary of n atoms (column vectors) and a data vector. OMP encodes data representations by selecting a small number k of the atoms, such that their linear combination best approximates the data vector. Its selection procedure only needs k successive iterations: in each iteration, the atom that can maximally reduce the residual error is selected. Such a greedy iterative solver, however, can be sensitive to data variations. The greedy selection process can amplify small differences in data and lead to large deviations in their representations.

In this section, we introduce a variant of OMP, named nonnegative OMP (NOMP), and show its improved stability in computing data representations. Throughout this work, we learn feature dictionaries using the spherical K-means algorithm (also known as "gain shape" vector quantization) [22] unless otherwise noted.

3.3.1 Nonnegative OMP

Given a nonnegative dictionary $\mathbf{D} \in \mathbb{R}^{m \times n}$ and a nonnegative data vector $\mathbf{x}$, NOMP finds an approximate solution to the following nonnegatively constrained problem:

$$\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2 \ \ s.t. \ \ \|\mathbf{z}\|_0 \leq k, \ z_i \geq 0 \ \ \forall i \tag{3.1}$$

That is, we would like to find sparse nonnegative coefficients $\mathbf{z} \in \mathbb{R}^n$ that can approximately reconstruct the data $\mathbf{x}$ using the corresponding k dictionary atoms, where k is a relatively small integer. NOMP iterates the following steps for up to k rounds:

1. Initialize the residual vector $\mathbf{r}^{(0)} = \mathbf{x}$ and round number $l = 1$. Select the atom $\mathbf{d}_{i_l}$ that has the highest positive correlation with the residual, $i_l = \arg\max_i \left\langle \mathbf{d}_i, \mathbf{r}^{(l-1)} \right\rangle$. Terminate if the largest correlation is less than or equal to zero.

2. Approximate the coefficients of the selected atoms by nonnegative least squares. $\mathbf{z}^{(l)} = \arg\min_{\mathbf{z}} \left\| \mathbf{x} - \sum_{h=1}^{l} \mathbf{d}_{i_h} z_{i_h} \right\|_2 \ \ s.t. \ z_{i_h} \geq 0$

3. Compute the new residual $\mathbf{r}^{(l)} = \mathbf{x} - \mathbf{D}\mathbf{z}^{(l)}$. Increment l by 1.

While following the high-level iterative structure of OMP, NOMP uses two special mechanisms. First, NOMP selects the atom that has the highest *positive* correlation with the residual, in contrast to OMP which considers both positive and negative correlations. NOMP may exit iterations early if there are no more atoms with positive correlations. Second, NOMP computes the sparse code using nonnegative least squares instead of conventional unconstrained least squares. Note that solving nonnegative least squares is considerably more expensive than solving its unconstrained variant. Empirically, we usually find it sufficient to approximate the solution by solv-

ing unconstrained least squares and truncating any resulting negative coefficients to zero.[1]

Given the structural similarity between NOMP and OMP, existing efficient OMP implementations, such as batch OMP [70], can easily be adopted by NOMP. These implementations usually exploit both the sparsity in coefficients and incremental updates between iterations. With a large dictionary and small k, the overall computation required is dominated by computing a single round of atom correlations $\mathbf{D}^T\mathbf{x}$. Note that the computation of least squares is not the dominating cost. In this case, NOMP has a running time comparable to other similar encoders, including OMP and soft-threshold encoders.

3.3.2 Nonnegative OMP as an Encoder

To use NOMP as an encoder, we need to ensure the nonnegativity of both dictionary and input.[2] We define a nonlinear mapping $S : \mathbb{R}^{\frac{m}{2}} \rightarrow \mathbb{R}^m_{\geq 0}$ that transforms the input data $\mathbf{x}^{in} \in \mathbb{R}^{\frac{m}{2}}$ into a nonnegative vector $\mathbf{x}$ that is double-sized, $S(\mathbf{x}^{in}) = [\max(0, \mathbf{x}^{in}), \max(0, -\mathbf{x}^{in})]$ where 0 denotes the zero vector with all its components being zero. For example, a length-2 data vector $[1, -1]$ is transformed to a length-4 vector $[1, 0, 0, 1]$. This transformation has been used in modeling the receptive fields in human vision systems [43]. Given nonnegative data, the K-means algorithm ensures a nonnegative dictionary will be learned.

[1] Although truncating the negative coefficients may result in the residual vector having nonzero correlations with the selected atoms, these correlations must be negative. The selected atoms thus will not be re-selected in later iterations, and NOMP's convergence property is not affected.

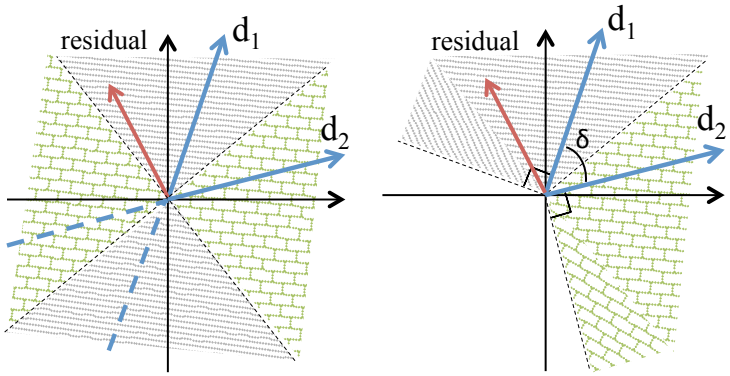
[2] Although pixel intensities are nonnegative, data preprocessing such as mean subtraction can generate negative values.

Table 3.1: A comparison in data dimensionality between unconstrained encoders and NOMP to compute a length-n feature vector from a length-$m/2$ data vector.

	$\mathbf{x}^{in}$	$\mathbf{x}$	$\mathbf{D}$	$\mathbf{z}$
UNCONSTRAINED ENCODERS	$\frac{m}{2}$	$\frac{m}{2}$	$\frac{m}{2} \times \frac{n}{2}$	$\frac{n}{2}$
NOMP	$\frac{m}{2}$	m	$m \times n$	n

Interestingly, prior research has observed that applying this sign splitting transformation with other unconstrained encoders leads to improved classification results [62, 22]. This splitting, however, is applied *after* the encoding step, for weighting the positive and negative feature vector values differently in a classifier. In this case, unconstrained encoders can be viewed as a form of nonnegative encoders with a dictionary $[\mathbf{D} \; -\mathbf{D}]$. NOMP generalizes this formulation by using a double-sized nonnegative dictionary that has no such special symmetric structures, and can be expected to be more powerful in classification. Nevertheless, we will see that the advantage of NOMP is beyond this generalization. Table 3.1 compares the data dimensionality in unconstrained encoders and NOMP.

Note that the nonnegative formulation allows only additive features, and cannot express cancellation between features efficiently. For image data, this limitation is less of a problem. The classical NMF result suggests that the nonnegativity constraint can lead to parts-based representations [52]. For deep, high-level representations, the nonnegativity constraint in fact is preferred, since nonzero entries in the representations correspond to activations of low-level features, and the cancellation between low-level features would be less meaningful.



(a) OMP (b) NOMP

Figure 3.1: Grey(green)-shaded area denotes where the residual vector sits that the algorithm would select $d_1(d_2)$ as the next atom. NOMP can tolerate larger variations in the residual and choose the same atom.

3.3.3 Stability of Nonnegative OMP Under Noisy Data

To use sparse representations for classification, it is important that the data representations are stable under expected small data variations. Unstable data representations can confuse supervised classifier training and result in poor classification performance. In this section, using noise as a proxy for small data variations, we assess the robustness of an encoder, and argue that a robust encoder is more stable under these data variations.

OMP is known to obey a local stability under noise [28]. That is, OMP can tolerate sufficiently small data noise and still find a sparse representation with the same support (the same nonzero entries). Figure 3.1(a) illustrates this local stability. Suppose we have two atoms $\mathbf{d}_1$ and $\mathbf{d}_2$ in the dictionary. Given the residual vector shown in the figure, OMP would select $\mathbf{d}_1$ as the next atom because the projection of the residual vector onto $\mathbf{d}_1$ is larger than its projection onto both $\mathbf{d}_2$ and $-\mathbf{d}_2$. This selection procedure allows the residual to be affected by small noise. If this

deviation is small enough such that the deviated residual does not fall out of the shaded area, the same atom $\mathbf{d}_1$ will still be selected by OMP. However, a slightly larger noise may cause OMP to select $-\mathbf{d}_2$ as the next atom, and subsequently the computed representation may differ by a large error due to a different support set.

In contrast, NOMP can tolerate a larger noise as illustrated in Figure 3.1(b). In NOMP, only the projections of the residual onto positive $\mathbf{d}_1$ and $\mathbf{d}_2$ are considered, giving a larger noise-tolerant area. Denoting the angle separating $\mathbf{d}_1$ and $\mathbf{d}_2$ as δ and considering the same residual vector, the noise-tolerant area for NOMP to choose $\mathbf{d}_1$ spans an angle of $\pi/2+\delta/2$, larger than OMP's $\pi/2$. This also suggests that NOMP's noise-tolerant region grows when the two dictionary atoms are further separated, while OMP's noise-tolerant region has a fixed size no matter how the angle between atoms is varied.

Formally, the following theorem shows that NOMP can tolerate sufficiently small noise in data and computes representations with the same support.

Theorem 7. *Suppose a data vector* $\mathbf{x}$ *has a nonnegative* k*-sparse representation* $\mathbf{z}$ *using a nonnegative dictionary* $\mathbf{D}$*, i.e.,* $\mathbf{x} = \mathbf{D}\mathbf{z}$*. Given a noisy data vector* $\mathbf{x} + \mathbf{n}$*, NOMP finds a sparse representation that has the same support as* $\mathbf{z}$ *if the noise* $\mathbf{n}$ *satisfies*

$$\frac{\|\mathbf{n}\|_2}{z_{min}} < \frac{\sqrt{2}}{2}(1 - \mu k) \tag{3.2}$$

where μ *is the coherence of the dictionary, or the maximum correlation between any two atoms in the dictionary, and* z_{min} *is the smallest nonzero entry in* $\mathbf{z}$*.*

Proof. We begin the proof by considering the first iteration in NOMP. Assuming the $\mathbf{z}$'s k nonzeros are located in the first k entries in descending order of magnitudes, for

NOMP to select a correct nonzero entry, we need

$$\max_{1\leq h\leq k} \langle \mathbf{x}+\mathbf{n}, \mathbf{d}_h\rangle > \max_{h>k} \langle \mathbf{x}+\mathbf{n}, \mathbf{d}_h\rangle \tag{3.3}$$

We can bound both sides of (3.3):

$$\begin{aligned} \langle \mathbf{x}+\mathbf{n}, \mathbf{d}_1\rangle &= z_1 + \sum_{i=2}^{k} z_i \langle \mathbf{d}_i, \mathbf{d}_1\rangle + \langle \mathbf{n}, \mathbf{d}_1\rangle \\ &\geq z_1 + \langle \mathbf{n}, \mathbf{d}_1\rangle \end{aligned} \tag{3.4}$$

$$\begin{aligned} \langle \mathbf{x}+\mathbf{n}, \mathbf{d}_h\rangle &= \sum_{i=1}^{k} z_i \langle \mathbf{d}_j, \mathbf{d}_h\rangle + \langle \mathbf{n}, \mathbf{d}_h\rangle \\ &\leq z_1\mu k + \langle \mathbf{n}, \mathbf{d}_h\rangle \end{aligned} \tag{3.5}$$

Combining (3.3)-(3.5) yields

$$\langle \mathbf{n}, \mathbf{d}_h\rangle - \langle \mathbf{n}, \mathbf{d}_1\rangle < z_1(1-\mu k) \tag{3.6}$$

Note that all the atoms are nonnegative. This allows us to further bound the left-hand side of (3.6).

$$\langle \mathbf{n}, \mathbf{d}_h\rangle - \langle \mathbf{n}, \mathbf{d}_1\rangle \leq \|\mathbf{n}\|_2 \|\mathbf{d}_h - \mathbf{d}_1\|_2 \leq \sqrt{2}\|\mathbf{n}\|_2 \tag{3.7}$$

Swapping (3.7) into (3.6) gives us a bound for the noise that NOMP selects a correct nonzero entry in the first iteration.

$$\frac{\|\mathbf{n}\|_2}{z_1} < \frac{\sqrt{2}}{2}(1-\mu k) \tag{3.8}$$

We can repeatedly apply the same procedure to derive bounds in later NOMP iterations. In the l-th iteration, we can find the following bound analogous to (3.8).

$$\frac{\|\mathbf{n}\|_2}{z_l} < \frac{\sqrt{2}}{2}(1-\mu k) \tag{3.9}$$

Therefore, satisfying the k conditions derived in the k NOMP iterations guarantees finding the correct support set, and (3.2) suffice to satisfy all k conditions. □

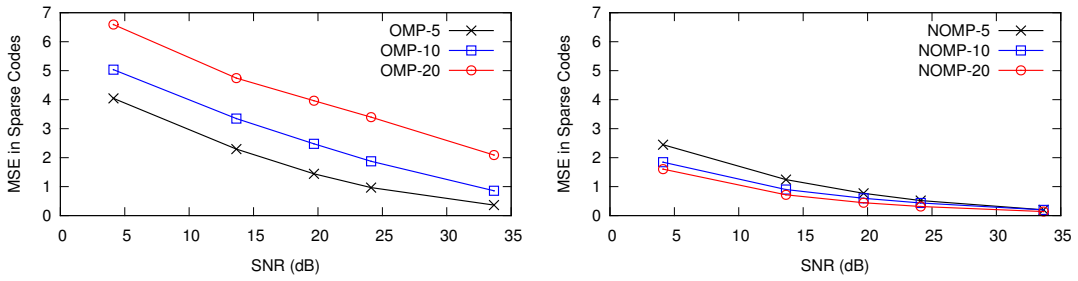
This upper bound for tolerable noise is larger than OMP's bound by a factor of $\sqrt{2}$, previously derived using the same proof technique [28]. We note that this bound is loose and empirically NOMP can tolerate even larger noise. We will provide empirical results in Section 3.4.

3.3.4 Improving Stability with Multiple Dictionaries

We have seen that NOMP enjoys a stronger stability than OMP. However, fundamentally, the stability of greedy pursuit algorithms is limited by the *coherence* of feature dictionaries, as strongly correlated atoms in the dictionary can cause more unstable atom selections. In practice, it is not easy to ensure all dictionary atoms to be equally separated, suggesting that NOMP's encoding will be particularly unstable to data related to strongly correlated atoms. A simple strategy to mitigate this problem is to make use of multiple separate dictionaries such that it is unlikely that a data input will have unstable representations across the majority of the dictionaries. One can then train multiple separate classifiers, each corresponding to one dictionary, and use a majority vote to combine the predictions.[3][4]

[3]Note that this technique does not improve the classification result when using the soft-threshold encoder. The soft-threshold encoder is stable regardless of the properties of dictionaries.

[4]In our implementation, we learn separate dictionaries by using different initializations to the K-means algorithm. The predictions reported in Section 3.6 are combined from 7 dictionaries and classifiers.



(a) OMP, MSE of sparse codes

(b) NOMP, MSE of sparse codes

Figure 3.2: Impact of noise on the stability of OMP and NOMP. NOMP is more stable under noise and shows less overfitting.

3.4 Empirical Validation of NOMP's Stability

In this section, we empirically validate NOMP's stability for data under noise, and data under variations. We use a dictionary learned from 6×6 images patches from the CIFAR-10 dataset.

For noisy data, we sample 10,000 image patches $\mathbf{x}$ from CIFAR-10, and generate 2,500 noisy versions $\mathbf{x}^*$ of each patch with different Gaussian noise. Both clean and noisy samples, $\mathbf{x}$ and $\mathbf{x}^*$, are encoded to sparse codes $\mathbf{z}$ and $\mathbf{z}^*$, respectively. We measure the mean-squared-error (MSE) of sparse codes $\|\mathbf{z}^* - \mathbf{z}\|_2$ to assess the stability of the encoders. The sparsity bound k of both OMP and NOMP are varied during the experiments, denoted as (N)OMP-k.

As shown in Figure 3.2(a), the MSE of sparse codes computed by OMP grows with a larger k. Using more atoms to approximate the input runs the risk of overfitting to data noise, and consequently leads to more unstable sparse codes. In contrast, in Figure 3.2(b), NOMP finds sparse codes with smaller MSE across all SNRs. In fact, the MSE even *drops* with a larger k due to the fact that NOMP would terminate

Table 3.2: Stability of the codes of grating images under rotations.

ANGLE	0	0.01π	0.02π	0.03π	0.04π
OMP-5	1.00	0.71	0.54	0.43	0.34
NOMP-20	1.00	0.92	0.80	0.68	0.57

by itself when no more additive atoms can be found, effectively reducing overfitting. This result suggests NOMP-20 potentially is a better encoder than OMP-5.

Next we test the stability of sparse codes for images of grating under small rotations. We generate 8,000 6×6 images of grating and rotate each image by some small angle.[5] For each pair of grating and its rotation, we compare the similarity between their sparse codes with the normalized correlation. As shown in Table 3.2, the codes computed by NOMP are more stable under small rotations while the codes computed by OMP quickly become very different.

3.5 A Multi-Layer Learning Framework for Classification with NOMP

We adopt a popular architecture that stacks multiple layers of convolutional feature encoders [54, 23]. At each layer, overlapping patches from the input feature maps are encoded using a feature dictionary. The computed representations are then pooled (max or average) over a small neighborhood to generate feature maps for further encoding in the next layer, or pooled over the whole image to form an image

[5]The grating is generated by $I(x, y) = b + a\sin(\omega(x\cos\theta + y\sin\theta - \phi))$ where ω is the spatial frequency, θ is the orientation, and ϕ is the phase [14]. We set $\omega = \{0.5, 1, 1.5, 2\}$ and $\phi = 0$ to π with a step size $\pi/20$.

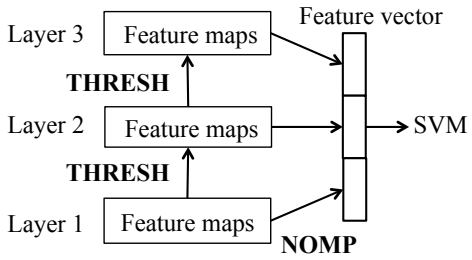


Figure 3.3: The learning architecture adopted in this work. Note that we use different encoding methods to compute sparse representations for classifier (NOMP) and for higher-level encoding (soft-threshold).

representation.

Standard preprocessing steps are applied on image data to generate data vectors for layer-1. These include mean subtraction, contrast normalization , and ZCA-whitening, followed by sign-splitting as described in Section 3.3.2.

However, unlike the popular architecture, we use different encoding methods to compute feature vectors for classification, and compute feature maps for higher-layer encoding, as illustrated in Figure 3.3. In particular, we use NOMP to compute sparse representations for classification, and a soft-threshold function to generate sparse codes for higher layers.[6] We do so because feature maps computed with NOMP are very sparse and are difficult to be further encoded efficiently, and therefore use the soft-threshold function for less-sparse representations. Empirically, we found that using a soft-threshold function that truncates 90% coefficients works well. Note that between layers, only a feature vector normalization step is performed. This makes the learning framework very simple as compared to other existing frameworks, which require some form of data whitening [23, 45].

For even faster computation, the nonnegativity constraint allows us to enforce a

[6]Note that this adds only very little computation, as $\mathbf{D}^T\mathbf{x}$ is already computed by NOMP.

sparse high-layer dictionary, given that high-layer inputs are sparse and only additive atoms are allowed. By exploiting the sparsity in computations, both training and encoding can be made significantly faster. A simple strategy to enforce sparse dictionaries is to drop entries with small values. We typically set atoms to have only 10% nonzeros for layer-2 and above. This shows no harm to classification accuracy in our experiments.

Finally, the representations computed at different layers are concatenated as a image feature vector for use in classification, for which we employ a linear classifier (L2-SVM).[7]

3.6 Validating NOMP with Classification

3.6.1 Performance on the CIFAR-10 Dataset

Single layer performance

We first evaluate NOMP using the full CIFAR-10 dataset with a single layer encoder. CIFAR-10 is a dataset with abundant labeled training samples (5000 for each class). We encode 6×6 patches, pool the sparse codes over the four quadrants of an image, and concatenate the four representations. For comparison purposes, this architecture is identical to those used in the literature [22, 39].[8] We compare NOMP with both OMP and the soft-threshold encoder, one of the best known encoders

[7]The feature vectors are standardized by rescaling the values to [0, 1] for each dimension. Note that this preserves the sparsity of feature vectors due to the nonnegativity.

[8]Accuracy is optimized over max- and average-pooling; generally, NOMP performs best with max-pooling, and the soft-threshold encoder with average pooling.

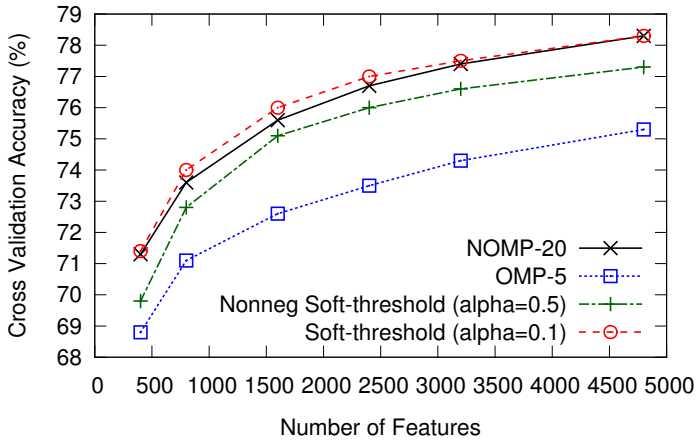


Figure 3.4: Single-layer classification accuracy on full CIFAR-10 with abundant training samples (5000 labeled samples per class).

for CIFAR-10. We choose k as 20 and 5 for NOMP and OMP, respectively, in the experiments.[9]

As shown in Figure 3.4, OMP achieves the worst accuracy despite being a sparse encoder. As a point for comparison, [22] uses ℓ_1-sparse coding and reports a 78.5% accuracy with 3200 features (compared to 74.5% achieved by OMP). This suggests that OMP, although very efficient, does not find good representations. With nonnegativity constraints, NOMP is able to approach the accuracy of ℓ_1-sparse coding (77.4%). This makes NOMP an attractive encoder, especially for its high computational efficiency as compared to ℓ_1-sparse coding.

Second, NOMP achieves accuracy comparable with the soft-threshold encoder in the full CIFAR-10 dataset. Classification accuracy with soft-threshold-encoded representations, however, is only competitive under a large amount of labeled training

[9]This choice is cross-validated over $k = \{1, 3, 5, 10, 20\}$. In general, choosing k is not difficult. For NOMP, overfitting is less of a problem, as the algorithm would terminate early by itself. As such, it is safe to use a large k, and this tends to improve performance. For OMP, setting k is trickier due to the trade-off between representational power and overfitting. Usually a small k (such as 5 or 10) works best for OMP. See Section 3.6.1.

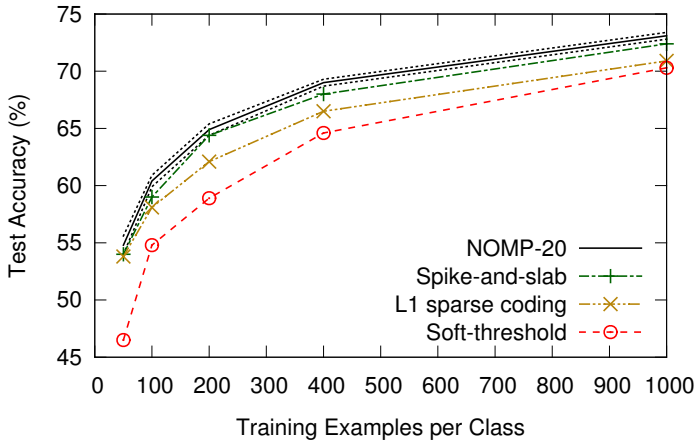


Figure 3.5: Single-layer classification accuracy on CIFAR-10 with fewer training samples (less than 1000 labeled samples per class). In this experiment, we use a dictionary of 3200 features. Only NOMP's standard error is shown for the readability of the figure.

samples [22]. In contrast, representations encoded by sparse encoders such as NOMP do not need as many samples. In Figure 3.5, we reduce the amount of labeled training samples in CIFAR-10, and compare NOMP to other encoders using the accuracy numbers reported by [39]. In this case, NOMP achieves the highest accuracy of the group when there are less than 1000 labeled samples per class, outperforming the soft-threshold encoder, the ℓ_1-sparse coding, and even the more sophisticated spike-and-slab sparse coding.

Quantifying the Impact of Nonnegative Training and Nonnegative Encoding

Having shown that NOMP not only delivers classification accuracy higher than OMP but also competitive with other well-known encoders, we seek to understand why NOMP performs well. NOMP differs with OMP in two ways: (1) Nonnegative training learns a more flexible dictionary by separating positive and negative channels

(see Section 3.3.2). (2) Nonnegative encoding enjoys a stronger stability than OMP (see Section 3.3.3). To measure individual impact of nonnegative training versus nonnegative encoding, we construct a dictionary $\mathbf{D_n}$ using a dictionary $\mathbf{D}$ trained from unconstrained K-means, and pair this dictionary with nonnegative encoding. $\mathbf{D_n}$ is constructed as follows to have a special symmetric structure as we do with unconstrained encoding:

$$\mathbf{D_n} = \begin{bmatrix} \max(0, \mathbf{D}) & \max(0, -\mathbf{D}) \\ \max(0, -\mathbf{D}) & \max(0, \mathbf{D}) \end{bmatrix} \tag{3.10}$$

In this experiment, we also include dictionaries formed by random numbers (R) and randomly selected patches (RP) for comparisons. The sparsity k of both OMP and NOMP for encoding is also varied to examine the impact of encoder stability on classification accuracy.[10]

We can make several observations from the results shown in Table 3.3. First, nonnegative encoding seems to contribute to most of the success of NOMP, and nonnegative training, in contrast, plays a minor role. We see higher classification accuracy even when nonnegative training is replaced by unconstrained training (76.3%). Across all dictionary training methods, encoding with NOMP-20 consistently improves classification accuracies by a significant margin.

Second, the stability of the encoders is strongly correlated with classification accuracy, and this explains why nonnegative encoding is particularly beneficial. In Section 3.4, we observed that a large k in OMP results in unstable representations. Correspondingly, we see OMP-20 delivers lower accuracy than OMP-5. In contrast,

[10]In these experiments we use features that have $3200 \times 4 = 12800$ dimensions for the SVM.

Table 3.3: Single-layer classification accuracy of CIFAR-10 using various training and encoding methods. We report accuracy from the 5-fold cross validation on the training set.

Training	Encoding			
	OMP-5	OMP-20	NOMP-5	NOMP-20
R	69.2	67.3	69.2	74.6
RP	72.9	71.6	74.6	77.1
Unsplit	74.5	73.6	74.9	76.3
Split	73.5	73.2	75.7	77.4

with NOMP, a larger k gives more stable representations, and we see the accuracy improves from NOMP-5 to NOMP-20. Note that these observations hold true regardless of the employed dictionary training method, suggesting that nonnegative encoding for improved encoder stability is of fundamental importance.

Multi-layer performance

We next examine NOMP's performance in a multi-layer, deep architecture as described in Section 3.5. For comparison, the settings of the architecture are identical to [23], where the authors stack multiple layers of soft-threshold encoders.[11]

As shown in Table 3.4, we can see that with NOMP, the accuracy can be improved effectively by simply adding more layers (78.0% to 81.4% in the full dataset, and 69.0% to 71.7% in the reduced dataset). We note that the only other known higher classification accuracy for the reduced CIFAR-10 is 72.6% [45], in which the accuracy is attained by exploiting view-invariant features rather than a better encoder design. Finally, exploiting multiple dictionaries can further improve the classification

[11]The patch sizes and features sizes are 6×6, 9×9, 15×15, and 3200, 6400, 6400 for the three layers, respectively.

Table 3.4: CIFAR-10 test accuracy in a multi-layer architecture.

	400 ex/class	Full data
OMP-5 (1 layer)	67.2 ± 0.3	75.2
OMP-5 (2 layers)	67.9 ± 0.4	76.4
NOMP-20 (1 layer)	69.0 ± 0.3	78.0
NOMP-20 (2 layers)	71.3 ± 0.4	80.9
NOMP-20 (3 layers)	71.7 ± 0.3	81.4
NOMP-20 (3 layers + multi dict)	72.2 ± 0.4	**82.9**
RF leraning (3 layers) [23]	70.7 ± 0.7	82
VIK [45]	**72.6 ± 0.9**	81.9

Table 3.5: Accuracy from 5-fold cross validation on full CIFAR-10 training set, using only representations constructed in layer-2. T denotes soft-threshold encoding, and NT denotes soft-thresholding with nonnegative sign splitting.

	Accuracy
Layer-1 (T) + Layer-2 (OMP)	67.3
Layer-1 (T) + Layer-2 (NOMP)	77.2
Layer-1 (NT) + Layer-2 (NOMP)	77.3

accuracy.

The accuracy improvement in stacked OMP, however, is relatively small. This may suggest that the nonnegative constraint is also advantageous for high-layer feature encoding. To evaluate the impact of nonnegative encoding in higher layers, we run another experiment that uses only the representations derived at layer-2 for classification.[12] In addition, we include a case where the nonnegative constraint is *only* enforced in layer-2. This allows us to isolate the impact of nonnegativity on layer-2.

Table 3.5 shows that two-layer OMP alone in fact achieves very poor accuracy (67.3%). Surprisingly, by only adding nonnegativity on layer-2, the accuracy can be

[12]Instead of concatenating representations found at all layers as described in Section 3.5.

drastically improved (77.2%) to almost the same as that in two-layer NOMP (77.3%). We hypothesize that the improvement is a result of avoiding unwanted cancellations between high-level features. A nonzero value in a high-level feature can be interpreted as the "presence" of the corresponding low-level feature. Therefore, cancellations between positive and negative values is less meaningful. Adding the nonnegativity constraint eliminates this possibility and again, prevents overfitting in the model.

We note that the current state-of-the-art accuracy of full CIFAR-10 is achieved by deep neural networks (e.g., [40]). The power of such methods, however, depends on the amount of available labeled training samples. As we will see in the next section, NOMP is very competitive when labeled training data is limited.

3.6.2 Performance on the CIFAR-100 Dataset

The strength of NOMP lies in its ability to tackle datasets with limited labeled training data. The CIFAR-100 dataset is one of such datasets: it has many more classes (100 classes), and fewer labeled training samples per class (500 samples for each class), as compared to the CIFAR-10 dataset (5000 samples per class). We use the same hyperparameters in this experiment as used in the CIFAR-10 experiments. As shown in Table 3.6, 3-layer NOMP achieves a very competitive accuracy (57.7%). Further, exploiting multiple dictionaries has a big impact. The accuracy can be largely improved (60.8%) and approaches the state-of-the-art accuracy achieved by maxout networks, an advanced extension of deep neural networks with dropout training.

Table 3.6: Classification accuracy of CIFAR-100.

	Test Acc.
OMP-5 (1 layer)	49.0
NOMP-20 (1 layer)	53.3
NOMP-20 (3 layers)	57.7
NOMP-20 (3 layers + multi dict)	60.8
Stochastic pooling [84]	57.5
Maxout [40]	**61.4**

3.6.3 Performance on the STL-10 Dataset

Finally, we evaluate NOMP on the STL-10 dataset, which features very few labeled training examples (100 examples for each of the 10 classes) and larger 96×96 images. Due to its relatively large image size, much prior research chose to downsample the images to 32×32. We examine NOMP's performance on both the downsampled and original-sized dataset.[13] The results are shown in Table 3.7.

First, we note that the classification accuracy follows similar trends as in CIFAR-10. With a single layer, NOMP achieves accuracy similar to ℓ_1-sparse coding. Using two layers of NOMP, the accuracy is also slightly better than that of three layers of stacked soft-threshold encoders.

Second, we found that image size has a huge impact on classification accuracy. Using the original image size, single-layer NOMP achieves an accuracy (64.6%) higher than all of the previously reported numbers. With 3 layers, NOMP achieves 67.5% accuracy, a new state-of-the-art accuracy. This result highlights the importance of

[13]For the downsampled dataset, we use the same setting for the multi-layer framework as in CIFAR-10. For the original dataset, we use 10×10, 19×19, and 38×38 patches for layer-1, layer-2 and layer-3, respectively. 2×2 max-pooling is inserted between layers. For both experiments, we use feature size 3200, 6400, and 6400 for the three layers, respectively.

Table 3.7: Classification accuracy of STL-10.

	Test Acc.
Thresh (1 layer, downsampled)	54.8 ± 0.4
OMP-5 (1 layer, downsampled)	58.1 ± 0.5
NOMP-20 (1 layer, downsampled)	59.0 ± 0.5
NOMP-20 (2 layers, downsampled)	60.4 ± 0.5
NOMP-20 (1 layer)	64.6 ± 0.6
NOMP-20 (2 layers)	67.0 ± 0.5
NOMP-20 (3 layers)	67.5 ± 0.5
NOMP-20 (3 layers + multi dict)	**67.9 ± 0.6**
Sparse coding (1 layer, downsampled) [22]	59.0 ± 0.8
RF leraning (3 layers, downsampled) [23]	60.1 ± 1
SPN [35]	62.3 ± 1
HMP [17]	64.5 ± 1

efficient and scalable training and robust encoding algorithms.

3.7 Summary

In this chapter, we have studied greedy sparse encoders for use in unsupervised sparse representation learning. We have found that the stability of OMP, known to be relatively weak, is the cause of its suboptimal classification accuracy. We have demonstrated that this issue can be largely alleviated by simply adding a nonnegativity constraint. The proposed NOMP encoder is not only very efficient, but also delivers competitive accuracy to other best known encoders, including deep neural networks, when the amount of labeled training samples is limited. This makes NOMP very attractive to building large-scale image classification systems.

Chapter 4

Medium Access Control for Multiuser MIMO Networks

In this chapter, we will discuss our second application: medium access control in multiuser MIMO networks. MIMO technology enables spatial multiplexing that the wireless channel bandwidth can be increased proportionally to the number of available transmit and receive antennas pairs. In the multiuser setting where each host station has one transmit antenna and the base station has many receive antennas, spatial multiplexing means that many host stations can transmit packets concurrently without resulting in packet collisions. This imposes new challenges in medium access control: the access pattern of concurrent transmissions must allow source identification and channel statistics estimation that are necessary for MIMO decoding.

We will see that sparse recovery allows us to identify transmitting host stations and estimate channel statistics using overlapping symbol sequences. More importantly, with sparse recovery, the transmitters only need to be loosely synchronized

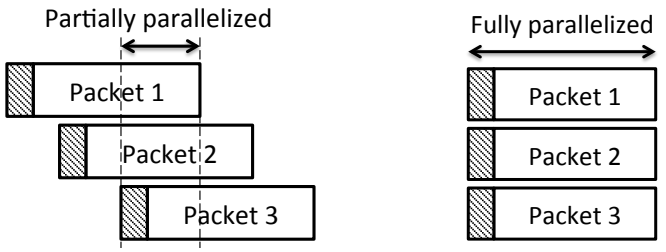
when transmitting the symbol sequences. Furthermore, we will show that the sparse recovery algorithm can be speeded up by leveraging the receive antenna diversity on the base station. The antenna diversity gives a "same-support" constraint to the recovery problem, as the received symbol sequences by different antennas share the same set of source senders.

4.1 Introduction

MIMO technologies can increase wireless channel capacity by exploiting the additional degrees of freedom created from multiple antennas. This capacity gain is particularly scalable in multiuser MIMO (MU-MIMO) [36], as the transmit antennas sit on geographically separated devices and offer a rich spatial diversity. In contrast, the spatial diversity in single-user MIMO is often limited by the co-located antennas installed on the same hardware platform.

In this chapter, we consider an MU-MIMO scenario in which an access point (AP) is equipped with multiple antennas, while every user has one antenna. We focus on the uplink case where multiple indoor MU-MIMO users, i.e., the "senders", send data packets concurrently to a multi-antenna AP. With MU-MIMO, one would expect that the data throughput can be increased by a factor of k, if k receive antennas are available on the AP, and sufficient spatial diversity exists among the k transmitters.

However, in practice, the realized throughput can be substantially less than that offered by the available degrees of freedom, due to the difficulty in estimating channel state information (CSI) from overlapping and mutually-interfered packets. Existing proposals of MU-MIMO systems (e.g., [76, 56]) allow proper channel estimation by



(a) Staggered access (b) Concurrent access

Figure 4.1: Two access strategies for multiuser MIMO networks. Shaded areas denote packet preambles. Staggered access has only partially parallelized data transmissions, resulting in low channel utilization. In contrast, concurrent access can realize MIMO capacity gain by fully parallelizing data transmissions.

avoiding preamble collisions, and suggest a staggered packet transmission pattern as shown in Figure 4.1(a). Such staggered transmission pattern only partially parallelizes packets in time, as each packet has to experience a separate access delay before transmission. As a result, the large cumulative delay seriously limits the throughput scalability to larger MIMO degrees of freedom. For instance, consider transmitting multiple 1500-byte packets in a 39Mbps PHY data rate. Given that each packet transmission spans 300μs, with an average access delay of 100μs [57], there will not be more than 3 packets transmitted concurrently. Although one can use frame aggregation [6] to allow a longer payload and amortize access overheads, such methods are not practical to delay sensitive applications such as VoIP or HTTP.

We argue that a more efficient approach to coordinating distributed senders is to launch multiple data transmissions simultaneously, thereby allowing the transmissions to be fully parallelized in time, as shown in Figure 4.1(b). We name this new access strategy *concurrent access*. To realize concurrent access, we propose a CSMA-based medium access control (MAC) design, named MIMO/CON ("MIMO with concurrent

access"), that has the following two features:

- MIMO/CON can derive accurate CSI using concurrent preambles that are both *loosely synchronized* and *not subject to centralized coordination*, suited to geographically separated senders with which symbol-level synchronization and communications are expensive.
- MIMO/CON can boost channel utilization by making use of collision packets. A "collision" in MU-MIMO means a set of concurrent transmissions containing more than k packets, while the MIMO AP has only k antennas. MIMO/CON mitigates the impact of these collisions by a novel scheme, called *delay packet decoding*, that can opportunistically decode collision packets at a later time. As a result, only a subset of packets involved in a collision needs to be retransmitted.

MIMO/CON exploits two important insights to realize these features. First, the amount of information, namely the CSI, we would like to collect is small. The channel impulse response typically contains only a few significant taps, i.e., delay paths, due to the short delay spread in an indoor environment. Second, allowing the overlapping concurrent preambles to be loosely synchronized and distributedly coordinated does not introduce additional CSI to be estimated. Instead, this will introduce additional unknowns that correspond to the candidate timing offsets and potential concurrent senders in the network (see Section 4.2). Solving the concurrent channel estimation problem thus can be viewed as a two-step process, in which the sparse set of senders and timing offsets can first be identified, and the channel responses can then be estimated.

MIMO/CON leverages the recent theory of compressive sensing [33] to solve the above sparse estimation problem. Compressive sensing shows that one can efficiently acquire sparse information without first knowing the locations of the nonzero entries. In other words, MIMO/CON can perform CSI estimation almost as if the concurrent preambles are transmitted with perfect synchronization from a set of known senders. Otherwise, the channel estimation will require an unpractically long preamble length.

We have prototyped MIMO/CON using software-defined radios, and performed evaluation through testbed experiments and simulations. Our evaluation reveals the following:

- The aggregated network throughput of MIMO/CON scales well with the number of available AP antennas. In particular, our simulation results suggest that MIMO/CON delivers 140% and 210% throughput gains over staggered access with a 5-antenna AP under PHY rates 13Mbps and 52Mbps, respectively (see Section 4.6.4).

- MIMO/CON's channel estimation scheme works over a wide range of SNRs. In particular, the estimated CSI can be used to decode MIMO packets with SNR as low as 5dB, the minimum required for MIMO packet transmissions. The decoded packet SNRs are comparable to those decoded using CSI estimated from interference-free, serially transmitted preambles (see Section 4.6.1).

- Compressive sensing is a natural fit for MIMO channel estimation. The antenna diversity presented by multiple receive antennas on a MIMO AP can be leveraged to speed up compressive sensing decoding. On our testbed, using 4 receive

antennas, the decoding algorithm can identify the correct nonzero entries in 1 to 2 iterations in the experiments (see Section 4.6.2).

In short, MIMO/CON achieves high and scalable MAC efficiency that can take full advantage of the available AP antennas. This is particularly important for future massive MIMO designs that allow many antennas to be installed on an AP (e.g., 802.11ac suggests up to 8 antennas on the AP, and an unlimited number of antennas scenario is depicted in [58] for cellular networks).

4.2 Channel Estimation and Exploitable Sparsity

4.2.1 Concurrent Access

Spatial multiplexing in MIMO systems uses distinct spatial signatures of transmit antennas to separate concurrent data transmissions. At the AP, the received signals can be viewed as a vector sitting in a high dimensional signal space, and each individual transmission will occupy a dimension defined by its associated spatial signature. By projecting the received signal onto proper subspaces, interference between individual transmissions can be eliminated, and individual data stream can be decoupled and decoded (for more details, see, e.g., [80]).

Computing the subspaces for decoding, however, requires estimating the channel state information (CSI). CSI is usually estimated by using a known preamble sequence preceding a data packet. To avoid mutual interference between the preambles, typically they are transmitted sequentially in single-user MIMO systems, as transmit antennas are co-located on the same hardware platform and can be easily

coordinated.

In multiuser MIMO systems, it is less clear how to perform channel estimation from overlapping packets. In [76], the authors show that channel estimation with overlapping packets is still possible as long as the preambles do not overlap. The authors have each packet transmission undergo a separate contention delay in order to avoid preamble collisions, giving a staggered transmission pattern (as shown in Figure 4.1(a)). The AP can then estimate the CSI by sequentially projecting the received preambles onto interference-free subspaces. As mentioned earlier, staggered data transmissions cannot utilize the channel efficiently. The k access delays in a set of k concurrent transmissions will be a major bottleneck for MIMO throughput.

MIMO/CON takes an opposite approach, named concurrent access, that explicitly overlaps multiple preambles in time, and exploits sparse recovery techniques for channel estimation. By launching multiple data transmissions simultaneously, the contention delay is paid only once for one set of concurrent transmissions (as shown in Figure 4.1(b)).

MIMO/CON realizes simultaneous transmissions from distributed senders using random-access-based MAC approaches. Assuming the backoff operations of the senders are in lockstep with each other, concurrent transmissions will naturally occur if the senders' transmission probabilities in a time slot are sufficiently high. Similar to 802.11 DCF, the transmission probability can be adjusted based on contention levels, i.e., observing the statistics of packet collisions. Since a larger number of AP antennas will allow more senders to transmit concurrently, the senders can gradually increase the transmission probability until hitting a collision. This also suggests that

under this scheme, the senders do not need to be aware of AP's MIMO capability to realize the capacity gain.

We note that the synchronization among senders in MIMO/CON refers to a loose one that a substantial synchronization offset between transmissions can be tolerated. We will discuss how a loose synchronization will impact channel estimation in Section 4.2.2, and how MIMO/CON can avoid the difficult symbol-level synchronization for channel estimation in Section 4.3.

4.2.2 Channel Estimation

The key assumption in concurrent access is that one can perform channel estimation using overlapping preamble sequences. Before formulating the concurrent channel estimation problem, let us first describe how CSI is estimated in a single-transmitter, single-receiver case.

CSI describes the distortion of a transmitted symbol in both amplitude and phase when passing through a channel. In a multipath environment, CSI can be modeled as a complex vector $\mathbf{h}$, where each entry represents the channel distortion along paths of a particular propagation delay. Denote the preamble as a length-m vector $\mathbf{d}$, $\mathbf{y}$ as the signal received at the receiver, and $\mathbf{n}$ as noise. Typically the CSI is estimated by solving (4.1), where the convolution models the multipath intersymbol interference.

$$\mathbf{y} = \mathbf{d} \otimes \mathbf{h} + \mathbf{n} \tag{4.1}$$

For clarity, we rewrite (4.1) into a matrix form with a circulant matrix $\mathbf{D}$ formed by

the preambles sequence.

$$\mathbf{y} = \mathbf{D}\mathbf{h} + \mathbf{n}, \text{ where } \mathbf{D} = \begin{bmatrix} d_1 & d_2 & \cdots & d_m \\ d_2 & d_3 & \cdots & d_1 \\ \vdots & \vdots & \ddots & \vdots \\ d_m & d_1 & \cdots & d_{m-1} \end{bmatrix} \tag{4.2}$$

Note that to solve (4.2), we will m to be larger than the length of $\mathbf{h}$ (i.e., the number of unknowns), which is determined by the path with the longest delay in the environment. Note that the delay spread in an indoor environment is typically small (30 to 60ns or even less [7, 37]). With a 20MHz bandwidth (25ns sampling interval), the length of $\mathbf{h}$ is no more than 3.

Next, let us consider a more general case where CSI is estimated from overlapping preambles, but the preambles are received with perfect synchronization (as depicted in Figure 4.2(a)) from a set of known senders. Suppose there are three concurrent senders, Sender 1, 2, and 3. The received signal can be modeled as a linear sum of the three preambles traveling through their corresponding channels.

$$\mathbf{y} = \mathbf{D}_1\mathbf{h}_1 + \mathbf{D}_2\mathbf{h}_2 + \mathbf{D}_3\mathbf{h}_3 = \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_2 & \mathbf{D}_3 \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} + \mathbf{n} \tag{4.3}$$

In this case, to solve (4.3) we need m to be about three times larger, larger than the total length of $\mathbf{h}_1$, $\mathbf{h}_2$, and $\mathbf{h}_3$.

MIMO/CON seeks to realize concurrent access with minimum centralized control, meaning that perfect symbol-level synchronization and the information on participating senders are both unavailable. As shown in Figure 4.2(b), loose synchronization

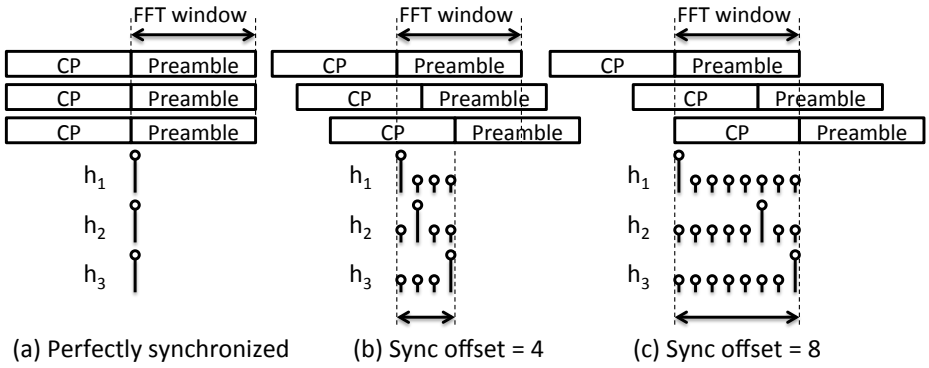


Figure 4.2: The number of unknowns in channel impulse response is proportional to the maximum synchronization offset.

among senders results in symbol misalignments as shown in Figure 4.2(b). This misalignment typically can be tolerated by exploiting the cyclic prefix (CP) structure in OFDM symbols, which is a repetition of the end of an OFDM symbol. Therefore, even if the misalignment exists, the receiver can still find a proper FFT window covering the same information of every symbol, as long as the CP is longer than the maximum synchronization offset.

In channel estimation, an FFT window unaligned with the beginning of a preamble will increase the number of unknowns in the CSI vector $\mathbf{h}_i$. The timing misalignment introduces an artificial delay to the preamble, and therefore adds additional potential preamble arrival time into $\mathbf{h}_i$. For example, in Figure 4.2(a), suppose the environment has only a single path. The length of $\mathbf{h}_i$ in Figure 4.2(a) would be only 1 and its value captures the channel distortion along this path. With a synchronization offset of 4 samples, as shown in Figure 4.2(b), the preambles will arrive at 4 potential time slots, and the length of $\mathbf{h}_i$ is increased to 4 to accommodate these 4 possibilities. Similarly, in Figure 4.2(c), a larger offset of 8 samples increases the length of $\mathbf{h}_i$ to 8. Note that although the number of unknowns grows linearly with the synchronization offset, the

number of nonzeros in $\mathbf{h}_i$ stays the same.

The number of unknowns in (4.3) will further increase if the identities of the senders are not known at the AP. In this case, we will need to include all potential n senders in the network into CSI estimation. This distributedly coordinated channel estimation problem with loosely synchronized preambles can be formulated as follows.

$$\mathbf{y} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_2 & \cdots & \mathbf{D}_n \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_n \end{bmatrix} + \mathbf{n} \tag{4.4}$$

Note that $\mathbf{h}_i$ is a $T_s \times 1$ vector, with T_s denoting the maximum synchronization offsets between transmissions. The total number of unknowns in (4.4) is nT_s.

Solving (4.4) by canonical least squares methods, however, will need the preamble length m to be greater than nT_s, which can be unpractical long. For example, a loose synchronization method, such as the reference broadcast [74], gives a 2μs accuracy. This accuracy translats to a maximum synchronization offset T_s equal to 80 samples under 20MHz bandwidth. Assume the network has $n = 100$ senders. This suggests that m has to be at least 8000, meaning a 200μs long preamble.

Note that the number of nonzero unknowns in (4.4) is small. Suppose that only 4 out of 100 senders participate in the transmission and each CSI vector has no more than 3 taps. We only need to solve for 12 unknowns, if the participating senders and the timing misalignments for the preambles were known. A 300ns long preamble would suffice to estimate the CSI in this case. This motivates us to exploit compressive sensing techniques for concurrent channel estimation.

4.3 Concurrent Multiuser CSI Estimation

As discussed in the previous section, the challenge in estimating CSI from overlapping preambles is that although the information we want to capture is small, there can be a huge number of unknowns. In this section, we will show that we can leverage ideas from compressive sensing to overcome the explosion of unknowns.

Compressive sensing [26] shows that a few random linear projections of a sparse vector will contain sufficient information for exact recovery. Formally, for an $n \times 1$ sparse vector $\mathbf{x}$ with k nonzero entries, one can form an $m \times 1$ sketch vector $\mathbf{y}$ by taking random linear projections of $\mathbf{x}$. $\mathbf{x}$ can then be reconstructed exactly from $\mathbf{y}$ with high probability, using an m as small as $O(k \log n)$, a small constant factor of the number of nonzeros k. In other words, by exploiting compressive sensing, MIMO/CON can perform concurrent channel estimation almost as if the timing misalignments and senders' identities are known apriori.

4.3.1 Random preamble sequences for CSI estimation

A natural way to form random projections of multiple CSI vectors is to use random preamble sequences. For simplicity, we use the preamble sequences of $\{1, -1\}$ drawn from Bernoulli distribution. To work with OFDM, we assume the number of OFDM subcarriers is equal to the preamble length m, and the preamble sequence is transmitted over the subcarriers.

We first write (4.4) in its frequency-domain form as we are interested in transmitting preambles and data over OFDM subcarriers. Denote the preamble sequence owned by sender i as a vector $\hat{\mathbf{a}}_i$, and $\hat{\mathbf{y}}$ as the signal vector received at the AP over the

subcarriers.[1] $\hat{\mathbf{y}}$ can be written as a linear combination of the preambles transmitted from the senders, where each subcarrier is a flat-fading channel.

$$\hat{\mathbf{y}} = \begin{bmatrix} diag(\hat{\mathbf{a}}_1) & diag(\hat{\mathbf{a}}_2) & \cdots & diag(\hat{\mathbf{a}}_n) \end{bmatrix} \begin{bmatrix} x_1\hat{\mathbf{h}}_1 \\ x_2\hat{\mathbf{h}}_2 \\ \vdots \\ x_n\hat{\mathbf{h}}_n \end{bmatrix} + \hat{\mathbf{n}} \tag{4.5}$$

Note that we use the "hat" notation to denote a frequency-domain representation. In addition, we use a {0,1} binary variable x_i to indicate whether Sender i is active within a total of n senders, given that the non-active senders do not contribute to the received signal.

As stated in the previous section, the channel response is sparse in its time-domain representation. Hence we can convert (4.5) into a sparse recovery problem by taking inverse Fourier transform on individual channel responses from each sender.

$$\hat{\mathbf{y}} = \mathbf{A}\mathbf{h} + \hat{\mathbf{n}} = \begin{bmatrix} \Phi_1\mathbf{F} & \Phi_2\mathbf{F} & \cdots & \Phi_n\mathbf{F} \end{bmatrix} \begin{bmatrix} x_1\mathbf{h}_1 \\ x_2\mathbf{h}_2 \\ \vdots \\ x_N\mathbf{h}_n \end{bmatrix} + \hat{\mathbf{n}} \tag{4.6}$$

where $\Phi_i = diag(\hat{\mathbf{a}}_i)$.

We are now ready to interpret (4.6) using compressive sensing. Note that the delay spread T_d is the same for every channel between the sender and the AP. Assume the number of active senders is k. $\mathbf{h}$ is then a T_dk-sparse vector of length mn. The received signal $\hat{\mathbf{y}}$ is a measurement vector with length m that takes random projections on

[1] To avoid confusion, we use a separate notation for the preamble sequence from (4.4).

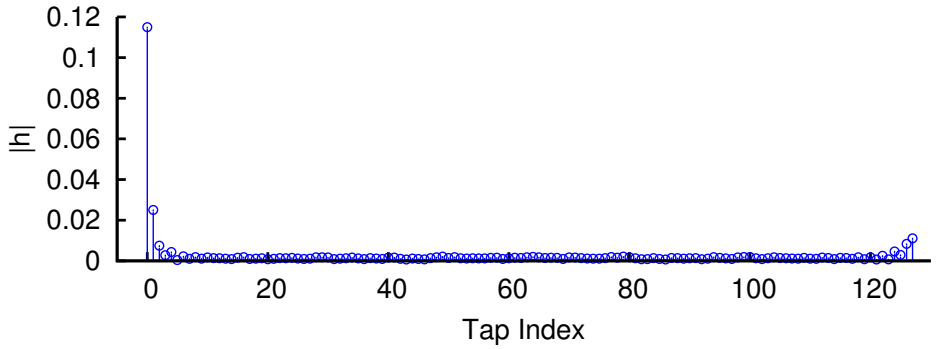


Figure 4.3: Channel impulse response measured with 6.25MHz bandwidth. A significant tap is observed at tap 0 with some energy leakage around.

h. By compressive sensing, to recover **h**, we only need m to be a small multiple of $T_d k$, which can be much less than mn. One can adjust m, the number of subcarriers or the FFT size in OFDM, for a longer measurement vector to accomodate different sparsity level. We note that the fact that m has to be larger than $T_d k$ means that the total number of samples used in concurrent channel estimation is slightly larger than that required in sequential preambles. Concurrent channel estimation thus requires a slightly higher power consumption.

Before we describe how to solve the sparse recovery problem in (4.6), there are a few points worth noting. First, the formulation can be thought of as a generalized form of CDMA that attempts to multiplex preambles without creating mutual interference. Traditional CDMA requires that the codes possessed by different senders to be orthogonal to each other. However, this assumes the worst case that all senders will transmit concurrently. Since we know the number of concurrent senders is bounded above by k, we can have a less constraining requirement that asks for only every subset of k codes to be orthogonal. This is exactly the formulation of compressive sensing that leads to a shorter code length.

Second, although the delay spread in an indoor environment is small and should contain only 1 or 2 significant taps, in practice the measured channel impulse response can have more nonzero taps due to leakage [81]. The leakage effect is a result of propagation delays that are not multiples of the sampling intervals. The energy of these delays leaks into every tap in the discretization process. Figure 4.3 shows a exemplar channel impulse response measured in an indoor environment with leakage. Fortunately, the leakage is concentrated around the most significant tap, and can be almost entirely captured by measuring a few additional neighboring taps.

4.3.2 CSI recovery with MIMO antenna diversity

MIMO decoding relies on per-packet spatial signatures, and therefore the sparse recovery of CSI must be solved within a packet time for decoding, which is around several hundreds of microseconds. However, sparse recovery is generally considered a computationally difficult problem. We will show that MIMO/CON can exploit the antenna diversity on the MIMO AP to relieve the computation burden, and shorten the decoding time.

Specifically, we note that the antenna diversity fits well with a popular class of greedy sparse recovery algorithms, such as Orthogonal Matching Pursuit (OMP) [79]. In greedy algorithms, an important step is to iteratively "guess" the locations of the nonzero unknowns. Once the locations are known, the linear system can be reduced to an over-constrained one, which is relatively easy to solve. As individual receive antennas obtain independent measurements, these measurements can naturally be combined to make the guessing more robust, which leads to a quicker convergence of

Algorithm 1 CoSaMP algorithm

Input: measurement vector $\hat{\mathbf{y}}$, sensing matrix $\mathbf{A}$, sparsity level $T_d k$, and noise tolerance *tol*

Output: CSI vector $\mathbf{h}^{(i)}$

1: $\mathbf{h}^{(0)} = 0$; $\mathbf{u} = \hat{\mathbf{y}}$; $i = 1$;

2: **while** $\|\mathbf{u}\|_2 > tol$ **do**

3: $\mathbf{p} = \mathbf{A}^T\mathbf{u}$ (Support estimation)

4: $\Omega = \text{supp}(\mathbf{p}_{\beta T_d k})$

5: $S = \Omega \cup \text{supp}(\mathbf{h}^{(i-1)})$ (Merge previous support)

6: $\mathbf{b}|_S = \mathbf{A}_S^{\dagger}\hat{\mathbf{y}}$ (Least squares to estimate signal)

7: $\mathbf{b}|_{S^c} = 0$

8: $\mathbf{h}^{(i)} = \mathbf{b}_{T_d k}$ (Prune estimate to be $T_d k$-sparse)

9: $\mathbf{u} = \hat{\mathbf{y}} - \mathbf{A}\mathbf{h}^{(i)}$ (Update signal residual)

10: $i = i + 1$

11: **end while**

the algorithm. Our decoding algorithm is an extension of CoSaMP [60], one of the most efficient algorithm for sparse recovery.

The CoSaMP algorithm

The pseudocode of the CoSaMP algorithm is shown in Algorithm 1. The core idea of CoSaMP is to identify the locations of the nonzero entries (the "support") in the solution vector, and refine the identified support set over the iterations. Due to the small number of nonzero entries in the solution, by only keeping the corresponding

atoms, the underdetermined linear system can be reduced to an over-constrained one, which can be solved by standard least squares methods.

The support set is estimated by evaluating the correlations between each atom and the measurement vector. Specifically, we compute a *proxy vector* $\mathbf{p} = \mathbf{A}^T\hat{\mathbf{y}}$ (as Line 3 in Algorithm 1), and include the atoms that have top correlation values into the support set. This estimation method can be understood as exploiting the *incoherence* of the measurement matrix. Given the atoms in $\mathbf{A}$ have low correlations with each other, we know its Gram matrix $\mathbf{A}^T\mathbf{A}$ is diagonally-dominant. Therefore the proxy vector $\mathbf{p} = \mathbf{A}^T\hat{\mathbf{y}} = \mathbf{A}^T\mathbf{A}\mathbf{h}$ has entries closely related to the true solution $\mathbf{h}$.

We note that we use a simple heuristic to exploit a special hierarchical structure in the solution vector $\mathbf{h}$ in (4.6) for support selection. As stated above, in general one can simply choose the atoms with the top $\beta T_d k$ correlation values into the support set (as Line 4 in Algorithm 1), with $\beta = 2$ as suggested by the original CoSaMP algorithm. In our application, the solution vector $\mathbf{h}$ has two levels of sparsity, wich allows us to perform support selection in two stages. The vector $\mathbf{h}$ can be divided into n length-m vectors, each representing the sparse channel response of a particular sender. Within the n senders, we expect k senders to be active, and the channel responses of the k active senders have no more than T_d nonzero entries each, summing to a total of $T_d k$ nonzero entries. Therefore, in the first stage of support selection, we select αk senders, and in the second stage, we choose βT_d nonzero entries from the selected senders. This approach helps avoid choosing nonzero entries spread in more than αk senders. In our implemntation, we set $\alpha = 1$ and $\beta = 2$.

Multi-antenna diversity

The k receive antennas on the AP offer k measurement vectors $\hat{\mathbf{y}}^{[1]}$ to $\hat{\mathbf{y}}^{[k]}$ of the same concurrently transmitted preambles. Each vector $\hat{\mathbf{y}}^{[i]}$ corresponds to a separate underdetermined system of linear equations, $\hat{\mathbf{y}}^{[i]} = \mathbf{A}\mathbf{h}^{[i]}$. Similar in spirit to the classic diversity combining schemes [80], we can "combine" the measurement vectors to improve the robustness of the recovery algorithm. However, the measurement vectors cannot be combined directly by simple additions.

To exploit this diversity, we make use of one important observation: the individual vector $\mathbf{h}^{[i]}$ share *the same* support despite they have different values. As described in Section 4.3, the locations of the nonzeros in $\mathbf{h}^{[i]}$ point to the identity of the senders as well as the timing offsets between individual transmissions. These factors remain unchanged across all co-located receive antennas.

This observation allows us to incorporate the antenna diversity into the CoSaMP algorithm with only one simple modification: we replace Line 3 of Algorithm 1 by the following:

$$\mathbf{p} = \frac{1}{k}\sum_{i=1}^{k} abs(\mathbf{A}^{\mathbf{T}}\mathbf{u}^{[i]}) \tag{4.7}$$

where $abs\,(\cdot)$ denotes taking element-wise absolute value of a vector and k is the number of AP antennas. Note that the k recovery problems are still solved independently using the original CoSaMP algorithm, but Line 3 is executed jointly.

Equation (4.7) can be viewed as noise reduction in estimating the proxy vector by taking an average over multiple noisy estimates. To see why (4.7) improves identifying the nonzero entries, let us illustrate with a simpler case where $\mathbf{h}^{[1]}$ has s nonzero entries, and the nonzero entries are the first s ones, $h_1^{[1]}$ to $h_s^{[1]}$. Denote the entries of

$\mathbf{A}^T\mathbf{A}$ as b_{ij} and note that the diagonal of $\mathbf{A}^T\mathbf{A}$ contains all 1's. We can expand the computation of the proxy vector $\mathbf{p}^{[1]} = \mathbf{A}^T\mathbf{A}\mathbf{h}^{[1]}$ as follows:

$$\begin{cases} \text{(nonzeros)} \;\; p_i^{[1]} = \left| h_i^{[1]} + \sum_{j=1, j\neq i}^{s} b_{ij} h_j^{[1]} \right| & \text{if } i = 1...s \\ \text{(zeros)} \qquad p_i^{[1]} = \left| \sum_{j=1}^{s} b_{ij} h_j^{[1]} \right| & \text{if } i > s \end{cases} \tag{4.8}$$

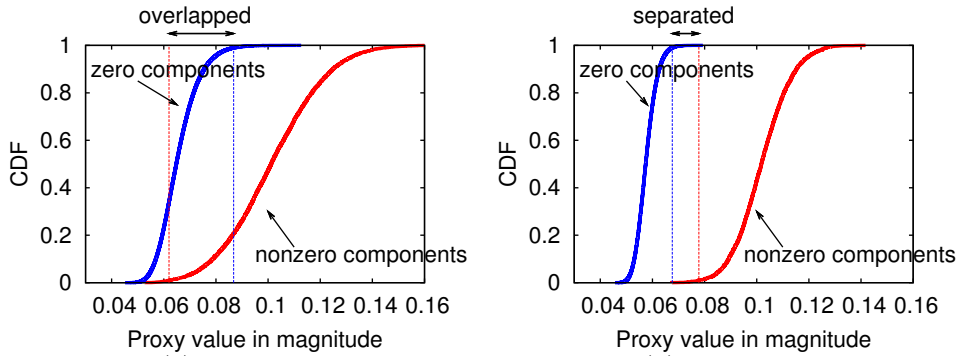
Note that as stated above, $\mathbf{A}^T\mathbf{A}$ is diagonally dominant, i.e. b_{ij} is small when $i \neq j$. Equation (4.8) suggests that the proxy $\mathbf{p}^{[1]}$ is equal to $abs(\mathbf{h}^{[1]})$ distorted by a small noise term $\sum_{j=1, j\neq i}^{s} b_{ij} h_j^{[1]}$. By approximating the noise term as a Gaussian variable, identifying the support from $\mathbf{p}^{[1]}$ is a detection problem where the entries in $\mathbf{p}^{[1]}$ corresponding to nonzeros in $\mathbf{h}^{[1]}$ are Gaussian variables with mean $h_i^{[1]}$, and the entries corresponding to zeros in $\mathbf{h}^{[1]}$ are half-Gaussian variables with mean zero. The misidentification rate of the nonzero entries is then determined by the size of the overlapping regions between the tails of the two Gaussian distributions.

Similarly, we can write the proxy vector estimated using $\mathbf{y}^{[2]}$ received at Antenna 2 as follows.

$$\begin{cases} \text{(nonzeros)} \;\; p_i^{[2]} = \left| h_i^{[2]} + \sum_{j=1, j\neq i}^{s} b_{ij} h_j^{[2]} \right| & \text{if } i = 1...s \\ \text{(zeros)} \qquad p_i^{[2]} = \left| \sum_{j=1}^{s} b_{ij} h_j^{[2]} \right| & \text{if } i > s \end{cases} \tag{4.9}$$

Note that in (4.8) and (4.9), b_{ij}'s are unchanged as they correspond to the preamble sequences. The only difference between the two linear systems is $\mathbf{h}^{[i]}$, which is the CSI of the channels between the receive antennas and Senders i.

We argue that the "noise terms" in (4.8) and (4.9), are independent. Therefore by summing the proxy vectors, we have a statistical gain that the noise variance can be reduced. In particular, assuming the noise term is a Gaussian variable, by taking



(a) 1 antenna (b) 3 antennas

Figure 4.4: Multi-antenna diversity improves the quality of support selection. Measurements from multiple antennas can help distinguish the locations of nonzero and zero variables.

the average of the k proxy vectors computed from the k antennas, the noise variance can be reduced in a rate of $O(k^{-1/2})$. The entries in the proxy vector quickly become very stable and robust.

The independence between the noise terms can be understood as follows. Note that the noise term is constituted by the CSI vector $\mathbf{h}^{[i]}$, which captures the multi-path signal attenuation and phase shift of the transmitted signals. Given the receive antennas are located on the same hardware platform and close to each other, the observed signal attenuation may not have too much diversity. However, their *phase shifts* can easily be very different. The wavelength of GHz waves is on the order of 10cm. It is thus reasonable to assume the nonzero entries in different $\mathbf{h}^{[i]}$ are independent variables.

We next conduct a numerical simulation to verify that the diversity in phase shift alone is sufficient to improve the recovery algorithm in identifying nonzero entries. In the simulation, we assume a scenario with 6 active senders out of a total of 100. Each channel response has one significant tap of magnitude 0.1 and 8 leakage taps

with magnitude 0.03. The signal attenuation observed at different antennas are set to the same value, and the phase shifts are sampled uniform randomly from [0,2π). The simulation is repeated 1000 times with different random preamble sequences and random locations of the nonzero entries.

We show the CDFs of entries in the proxy vectors that correspond to nonzero and zero unknowns, respectively. If the two CDFs are similar, identifying nonzeros is unlikely to be successful. On the other hand, if the two CDFs are much separated, identification is easy. Figure 4.4(a) shows the CDFs computed using measurements from one antenna. It can be seen that there is a non-negligible overlapping between the two CDFs. This means that although we can identify most of the nonzeros correctly, some nonzeros will not be selected due to their smaller values in the proxy vector. We will need to rely on additional iterations in the algorithm to further identify those entries. In contrast, in Figure 4.4(b) when the proxy is computed by using measurements from 3 antennas, the two CDFs are much separated, meaning that identifying nonzeros is much easier.

Computational complexity

The computational complexity of the CoSaMP algorithm is dominated by the support estimation step. With a naive matrix-vector multiplication, the computational complexity of $\mathbf{A}^T\hat{\mathbf{y}}$ is $O(nm^2)$. However, $\mathbf{A}$ is formed by multiple DFT matrices, and we can compute the proxy vector in blocks to exploit fast computation of DFT

transforms as follows.

$$\mathbf{p} = \mathbf{A}^T\hat{\mathbf{y}} = \begin{bmatrix} \mathbf{F}^{-1}\mathbf{\Phi}_1^T\hat{\mathbf{y}}_1 \\ \mathbf{F}^{-1}\mathbf{\Phi}_2^T\hat{\mathbf{y}}_2 \\ \vdots \\ \mathbf{F}^{-1}\mathbf{\Phi}_n^T\hat{\mathbf{y}}_n \end{bmatrix} \tag{4.10}$$

Note that $\hat{\mathbf{y}}_i$ is the i-th $m \times 1$ block in $\hat{\mathbf{y}}$. Given that $\mathbf{\Phi}_i^T$ is a diagonal matrix, computing (4.10) amounts to n DFT transforms, and the overall complexity is $O(nm \log m)$ with FFT. The other computationally extensive component in the algorithm is solving the least squares problems, whose complexity is $O(mT_d k)$.

4.4 Maximizing Channel Utilization

Beyond concurrent channel estimation, the MIMO/CON MAC layer needs to control the number of concurrent senders to maximize channel utilization. Suppose the AP has k receive antennas. Ideally we would like to ensure that there are always k senders to transmit concurrently. However, because random access by distributed senders inevitably leads to fluctuations between channel underutilizing (less than k senders) and channel overbooking (collisions), this problem cannot be generally solved without exchanging information between distributed senders.

Instead, MIMO/CON mitigates the channel utilization problem by *delay packet decoding*, which allows momentarily channel overbooking. The opportunity arises from two observations: first, concurrent channel estimation is not constrained by the number of available antennas on the AP. That is, with a proper preamble size, MIMO/CON can learn the sender identities and the associated CSI even in a MIMO

collision. Second, the MAC layer normally retransmits packets lost in a collision at a later time. Therefore, MIMO/CON can exploit the correctly received retransmissions to opportunistically decode packets lost in previous collisions.

To illustrate the idea, consider a simple scenario that the AP has two antennas, and at time t_1, three senders transmit packets p_1, p_2, and p_3 concurrently. Thus the AP receives:

$$\mathbf{y} = \mathbf{h}_1 p_1 + \mathbf{h}_2 p_2 + \mathbf{h}_3 p_3 \tag{4.11}$$

Since the AP only has two degrees of freedom, the AP cannot decode this set of concurrent transmissions. However, $\mathbf{h}_1$, $\mathbf{h}_2$, and $\mathbf{h}_3$ can still be estimated.

Suppose p_3 is retransmitted and received correctly at a later time t_2. Given that $\mathbf{h}_3$ has been learned in the previous collision, we can regenerate $\mathbf{h}_3 p_3$, and remove it from the linear system.

$$\mathbf{y} - \mathbf{h}_3 p_3 = \mathbf{h}_1 p_1 + \mathbf{h}_2 p_2 \tag{4.12}$$

Since this equation has only two unknowns left, we can proceed to decode p_1 and p_2.

MIMO/CON's approach to setting the transmission probability of every sender is similar to 802.11 DCF: when a sender sees a transmission opportunity, it tosses a coin to determine whether it will begin a transmission. If a collision occurs, the transmission probability is reduced to avoid future collisions. The classic additive increase multiplicative decrease (AIMD) control principle, for example, can be used to probe for optimal transmission probability and achieve fairness among senders.

4.5 Discussion

In this section, we discuss several practical issues in MIMO/CON.

(a) Hidden terminal: As in traditional CSMA, MIMO/CON avoids collisions through carrier sensing; therefore it also suffers from the hidden terminal problem that hidden senders cannot be detected. In addition to collisions, this problem may result in asynchronous concurrent transmissions that preambles are not overlapped as expected. When asynchronous concurrent transmissions happen without exceeding MIMO degree-of-freedom, one can apply techniques in staggered access, e.g., the chain decoding approach [76], to separate and decode data streams. On the other hand, when the contention level is high and collisions are likely, one will need to use RTS/CTS handshakes to contain the traffic. Interestingly, one can easily envision that the concurrent preambles can be a good primitive for building efficient concurrent RTS. A full design of the concurrent RTS, however, is beyond the scope of this work and left as future work.

(b) Frequency and time synchronization: Frequency and time synchronization is generally required in existing MU-MIMO systems, e.g., [56]. Since MIMO/CON does not impose more stringent requirements on synchronization accuracy, practical synchronization techniques in prior MU-MIMO systems are applicable. For frequency synchronization, the distributed senders can use the frequency of the AP's oscillator as a reference for synchronization. Since hardware oscillators have relatively stable frequency, frequency synchronization does not need to be performed

too frequently. For time synchronization, since symbol-level synchronization is not required in MIMO/CON, one can use coarse-grain time synchronization techniques such as the reference broadcast method [74]. Small timing misalignments between concurrent preambles can be tolerated by using the cyclic prefix structure in OFDM symbols as a guard interval, as discussed in Section 4.2.2. One can adjust the CP length to accomodate a larger synchronization error, and scale the data length accordingly to maintain the same CP overhead percentage.

Lastly, we note that although throughout the chapter, we assume that each sender has one antenna, the MIMO/CON design can easily be generalized to a multi-antenna sender case by having each sender operate as multiple single-antenna senders.

4.6 Performance Evaluation

We have implemented MIMO/CON on software-defined radios. We use the USRP-N200 boards with WBX daughterboards, and drive them with the UHD software [1]. The radios operate with a center frequency at 916MHz and a 6.25MHz bandwidth. The testbed is shown in Figure 4.5. In the testbed experiments, we focus on evaluating the performance of concurrent channel estimation. For delay packet decoding, there exists extensive empirical studies of interference cancellation techniques in the literature, e.g., [38][76]. Therefore we focus our evaluation of delay packet decoding on its performance gains in overall throughput.

In our implementation of concurrent preambles, we did not use the DC subcarrier to send the preamble sequence. This is to avoid unwanted DC offsets in the wire-

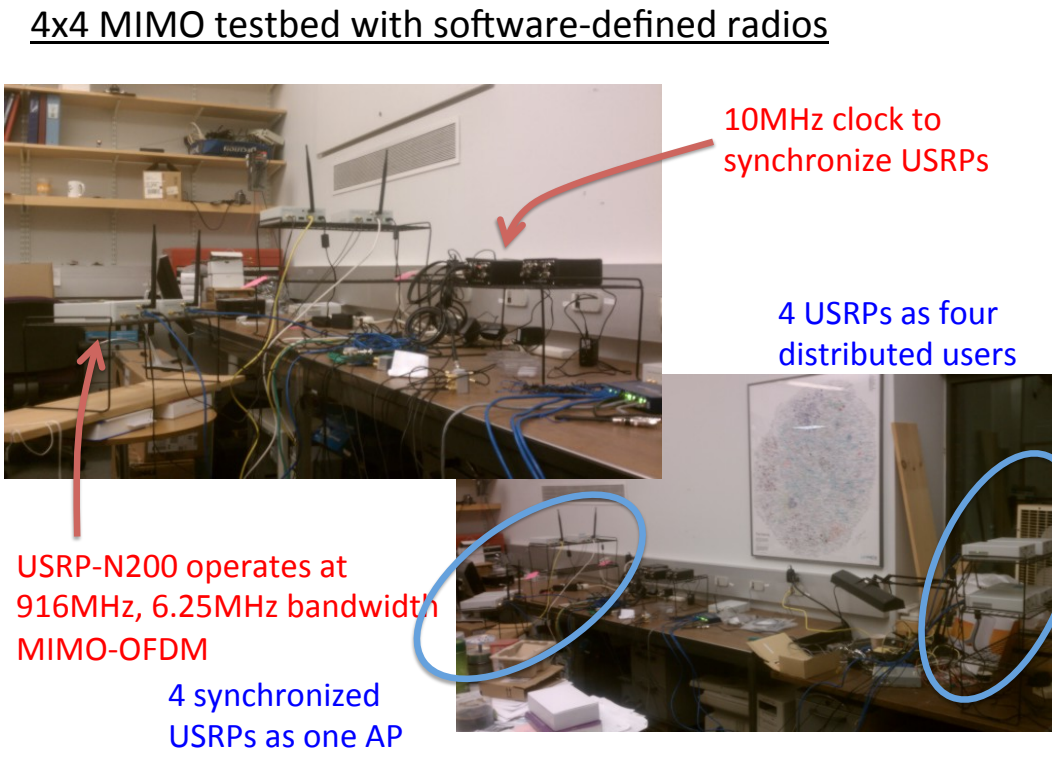


Figure 4.5: Software-defined Radio Testbed

less transceivers, which will shift the channel response by a nonzero constant, and eliminate the expected sparsity. Apart from the DC subcarrier, all other subcarriers are used for sending preamble sequences in order to correctly measure the channel response.

4.6.1 MIMO decoding performance with concurrent preambles

We use a 4×4 MIMO scenario to evaluate the performance of concurrent channel estimation in a lab environment. We compare the CSI estimated from concurrent preambles to a baseline case where preambles are transmitted sequentially and without mutual interference. We assume there are 100 senders but only 4 of them transmit at any given time. The distance between the transmitters and the receivers is set to be around 2 to 3 meters. We vary this distance as well as the transmission power to evaluate channel estimation under various SNRs.

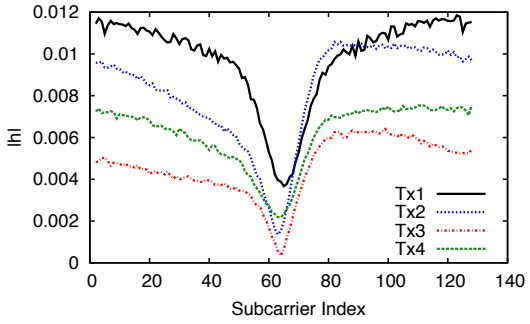
For the baseline scheme that estimates CSI using sequential preambles, we apply the standard least squares method [81].

$$\hat{\mathbf{h}} = \Phi^{-1}\hat{\mathbf{y}} \tag{4.13}$$

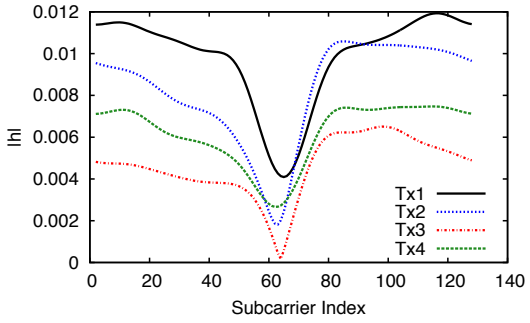
Recall that $\Phi = diag(\mathbf{a}_i)$ and $\mathbf{a}_i$ is the preamble sequence.

In the experiments, the senders transmit sequential and concurrent preambles followed immediately by data payloads. We use the CSI estimated from sequential and concurrent preambles, respectively, to decode the MIMO payloads, and compare their decoding performance. We use the standard zero-forcing method with successive interference cancellation [80] for MIMO decoding. The FFT sizes of both preamble and data symbols are set to 128 points. We repeat each experiment 300 times, and in each time, a different set of random preamble sequences is used.

We first examine the difference between the CSI estimated from sequential preambles and the CSI estimated from concurrent preambles. Figure 4.6 shows an example



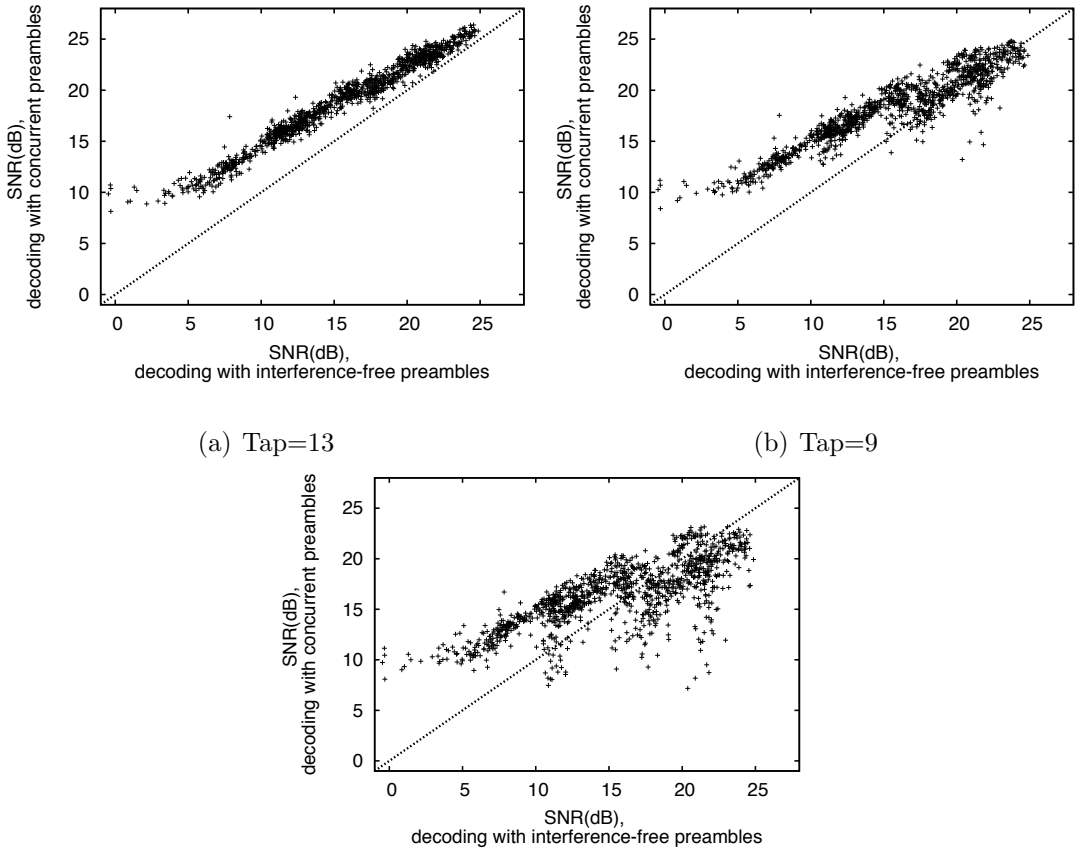
(a) CSI measured with interference-free preambles



(b) CSI measured with concurrent preambles

Figure 4.6: Comparison of the frequency domain CSI measured from interference-free preambles and concurrent preambles.

of the estimated channel frequency response on every subcarrier. The four curves in the figure are the channel responses from the four senders to a single receive antenna. It can be seen that the channel response estimated from sequential preambles are less smooth across subcarriers (Figure 4.6(a)). This is because the channel responses on individual subcarriers are estimated independently. In contrast, the channel response estimated from concurrent preambles appears a lot smoother (Figure 4.6(b)), reflecting the underlying assumption that the time domain channel response has only a few significant taps. Despite this difference in smoothness, the curves in both figures have values fairly close to each other, suggesting that concurrent channel estimation is able



(a) Tap=13

(b) Tap=9

(c) Tap=7

Figure 4.7: MIMO decoding performance using CSI estimated from concurrent preambles in 4×4 MIMO. Taking 13 taps is sufficient for reconstructing accurate CSI. Using fewer taps results in a degradation in decoding performance, especially when the signal SNR is high.

to obtain similar CSI to sequential channel estimation.

Next, we examine the MIMO decoding performance using the CSI estimated from concurrent preambles. We decode the 4 overlapping MIMO data transmissions using two different CSI, one estimated from sequential preambles, and the other from concurrent preambles. This allows us to obtain and compare the two different decoding SNRs. We then use a scatter plot to show all pairs of the decoding SNRs in the experiments, as shown in Figure 4.7. Note that in this figure, points above the

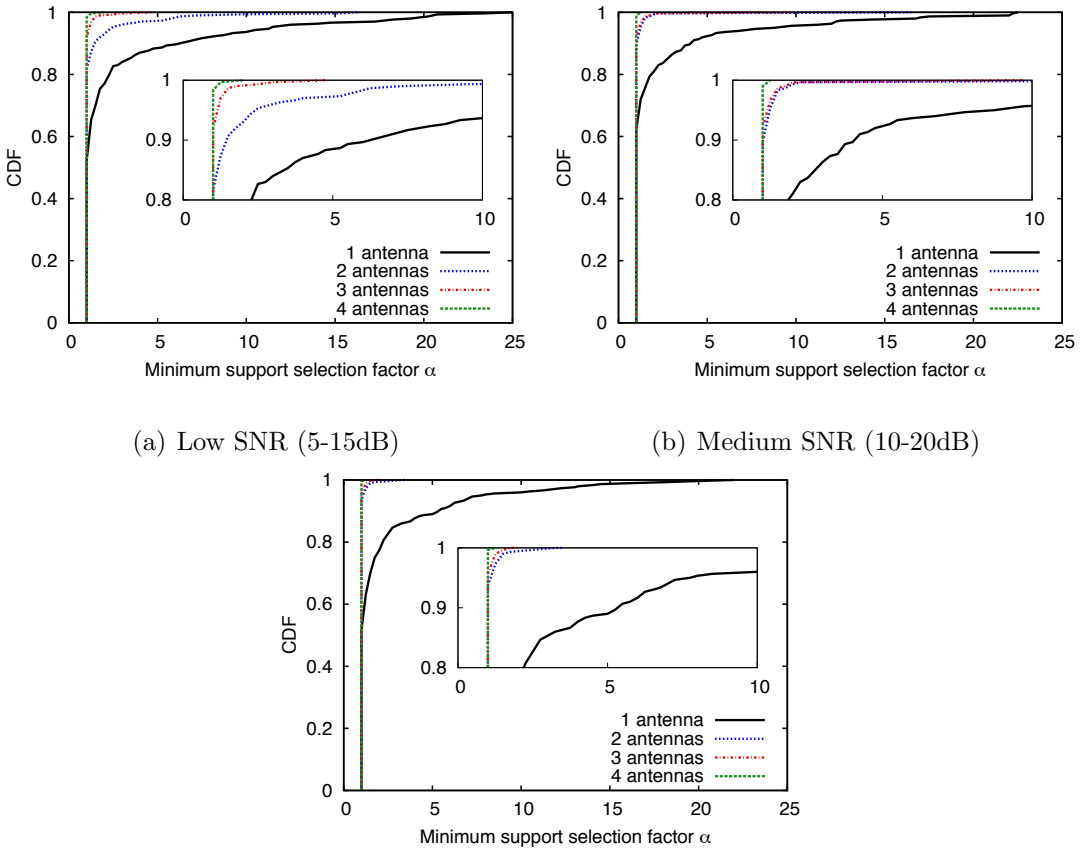
45-degree line mean that the MIMO decoding with CSI from concurrent preambles gives higher decoding SNRs. In this experiment, we also vary the expected number of nonzeros in the estimated CSI.

We can make several observations with the experimental results. First, with a sufficient number of nonzero entries in CSI, such as 13 taps (6 on each side of the most significant tap) in Figure 4.7(a), the decoding performance with concurrent preambles is better than with sequential preambles. This is because imposing the sparsity constraints helps filter out noise in channel estimation. A nonzero value appearing in a large-delay tap, which should have no response, is automatically suppressed during sparse signal recovery.

Second, allowing fewer nonzero entries in the CSI can result in a degradation in decoding SNR (Figure 4.7(b) and (c)) due to a less accurate CSI. However, when the signal SNR is low, the number of nonzero entries is less critical because the accuracy of CSI estimation is already limited by environmental noise.

4.6.2 Impact of antenna diversity in improving decoding efficiency

In this section, we examine the benefits of antenna diversity to sparse recovery algorithms using testbed data. As discussed in Section 4.3.2, incorporating measurements from different antennas can make the proxy vector robust, and this will facilitate support selection. To simplify our discussion, we focus on the first stage in support selection, which is to identify the active senders, in the two-stage approach discussed in Section 4.3.2.



(a) Low SNR (5-15dB)

(b) Medium SNR (10-20dB)

(c) High SNR (15-25dB)

Figure 4.8: Impact of antenna diversity. By incorporating just a few measurements from different antennas, one can estimate CSI from concurrent preambles in only one iteration of the decoding algorithm. Each plot includes a blown-up subplot to show details of CDF for α near 1.

We expect the proxy vector computed with antenna diversity has entries clearly separating the active and non-active senders. For each proxy vector, we measure the minimum α so that the top αk elements in the proxy vector include all active senders.[2] The minimum value of α is 1, meaning that the top k components in the proxy vector correspond exactly to the k active senders. A larger α indicates that

[2]The proxy vector here refers to the one used in the first stage to select active senders.

the proxy vector is more noisy, and support selection is more difficult.

Figure 4.8 shows the CDF of the computed α values in the experiments with different numbers of receive antennas and different SNRs. We can see that the proxy vector computed using multiple receive antennas quickly become very robust, with most of the α have a value of 1 . In the medium and high SNR case, using two receive datas, over 90% of the experiments have $\alpha = 1$, meaning that over 90% of the experiments, the active senders can be identified correctly in one iteration. With four receive antennas, almost all experiments have $\alpha = 1$. Similar trends can be observed in the low SNR case. With more receive antennas, a larger portion of α have values closer to 1, though in a slower rate.

4.6.3 FFT size of concurrent preambles

With a larger MIMO system with a higher degrees of freedom, in order to maintain a sufficient amount of measurements, we need to use a larger FFT size for the concurrent preambles. We use a simulation to examine the FFT size required under different number of concurrent senders. The simulation setting is largely the same as that in Section 4.3.2 with one difference. In this experiment, we allow 13 nonzero taps in the simulation, as suggested by the testbed results in Section 4.6.1. We test two FFT sizes, 128 and 256, and see the number of concurrent senders each FFT size can support for concurrent channel estimation.

As shown in Figure 4.9, the two FFT sizes can support up to 4 and 8 concurrent senders, respectively, with 100% success rate in sparse recovery. If we further exploit the antenna diversity articulated in this chapter, the same FFT size can support a

larger number of concurrent senders. In this case, the two FFT size can support up to 7 and 14 concurrent senders, respectively. In Figure 4.9, the vertical dotted lines (denoted as the "limit") shows the number of concurrent senders that can be supported by an FFT size, if the locations of the nonzeros are known. We note that in the 4-antenna case, the number of concurrent senders we can support is close to this limit (shown as vertical dotted line in Figure 4.9). This suggests that the additional overhead used to solve for the locations of the nonzeros is quite small.

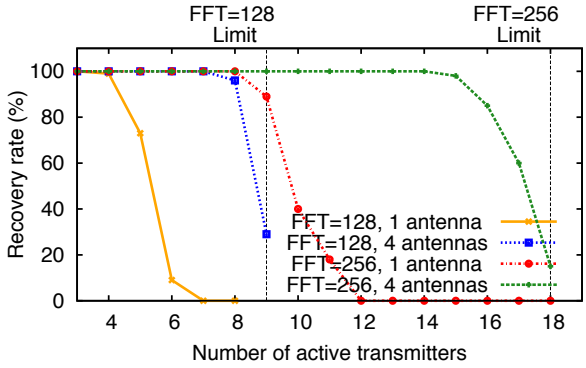
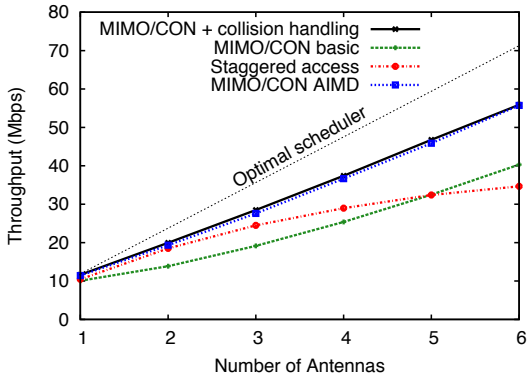


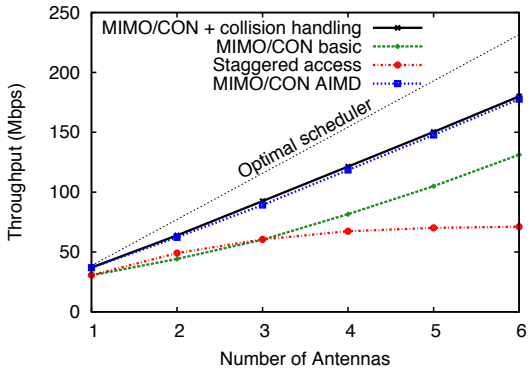
Figure 4.9: Length of concurrent preambles. Vertical dotted lines indicates the fundamental limit on the active senders that a particular FFT size can support.

4.6.4 Throughput improvement

In this section, we investigate the throughput improvement with MIMO/CON. We study MIMO/CON's throughput performance with software simulators. Due to hardware speed limitations, although we can implement concurrent channel estimation on software-defined radios, the current hardware system we have in lab cannot run fast enough to support carrier sensing and real-time concurrent preamble decoding for a large number of senders. We implemented an event-driven simulator with standard 802.11n parameters, including 28μs DIFS, 10μs SIFS, and 9μs slot time. In



(a) PHY data rate 13Mbps



(b) PHY data rate 52Mbps

Figure 4.10: Throughput of MIMO/CON with 20 nodes.

addition, we assume a standard 1500-byte data packet size and a 14-byte ACK packet size.

We compare MIMO/CON with SAM [76], a staggered access design for MU-MIMO systems. We first conduct an experiment assuming an environment with 20 senders that always have data packets to send. The senders are assumed to use the same PHY data rates. We assess MIMO/CON's performance under PHY data rates 13Mbps and 52Mbps, representing the low and high SNR regimes, respectively.

In the results shown in Figure 4.10, we first note that staggered access performs

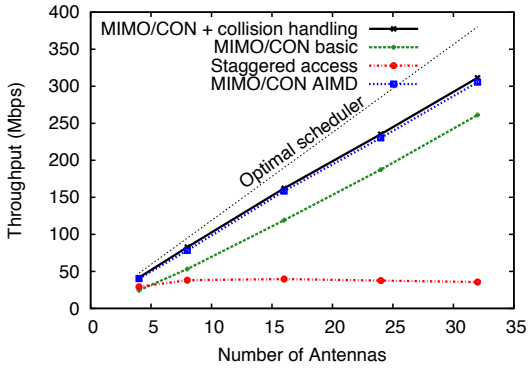


Figure 4.11: Throughput of MIMO/CON with 100 nodes and 13Mbps PHY data rate.

well when the number of AP antennas is small (and thus the number of expected concurrent senders is small), but the throughput quickly saturates when the number of AP antennas increases. This saturation is due to the serialized access delays of concurrent packets in staggered access. Given an average backoff period of 10 slots, the maximum number of overlapping packets in staggered access is upper-bounded by 8.5 and 2.7 packets under 13Mbps and 52Mbps data rates, respectively. No further throughput improvement is possible when the number of receive antennas is beyond this bound.

Second, MIMO/CON without delay packet decoding, denoted as "MIMO/CON basic", shows a good throughput scalability to a large number of AP antennas. The concurrent packets can be fully parallelized when channel estimation can be performed using overlapping preambles. However, MIMO/CON basic performs worse than staggered access when the number of AP antenna is small, due to the loss of efficiency from packet collisions. By employing delay packet decoding, denoted as "MIMO/CON + collision handling", the throughput achieved by MIMO/CON can be further improved, and is higher than that in staggered access.

Third, to simplify simulation, we simulate the above MIMO/CON schemes assuming a fixed transmission probability at the sender, and set the probability to the optimal value. We relax this assumption by adding the AIMD control to MIMO/CON to adaptively adjust the transmission probability. The results show that the AIMD control is effective in adapting the transmission probability, and MIMO/CON is able to deliver similar throughput performance. Overall, with 5 AP antennas, MIMO/CON can improve the throughput of staggered access by 140% under 13Mbps data rate, and a larger improvement of 210% under a higher 52Mbps data rate.

Finally, to understand the scalability of MIMO/CON, we conduct a larger scale simulation with 100 senders and up to 32 receive antennas on the AP. In Figure 4.11, MIMO/CON shows a good scalability to a larger amount of AP antennas. In contrast, staggered access cannot make use of the additional AP antennas and delivers a limited throughput.

4.7 Related Work

MIMO/CON is closely related to and is built on prior research of practical MU-MIMO systems [76][56]. We share the same goal with these works that we design a WiFi-like CSMA MAC scheme for MU-MIMO. As future MIMO designs are expected to see a substantial increase in the number of available antennas, MIMO/CON further addresses the scalability issue in MU-MIMO MAC designs.

Channel sparsity has long been exploited in channel estimation (see [9] for a nice review.) In particular, [68] makes use of random preamble sequences in channel estimation, similar to what we have proposed in MIMO/CON. However, MIMO/CON

stands out from these works in two ways. First, previous works assume the transmissions are perfectly synchronized and fully coordinated with known sender identities. MIMO/CON instead tackles the loosely synchronized WiFi environment, and leverages compressive sensing to efficiently estimate timing misalignments and senders identities. Second, previous works focus on PHY-layer improvements such as a enhanced demodulation performance and a reduction in pilot length. In contrast, MIMO/CON demonstrates that significant throughput gains can be realized at the MAC level, and compressive sensing techniques are of particular importance to MU-MIMO networking.

The MIMO/CON approach is closely related to CDMA-based channel contention approaches in which orthogonal codes are assigned to transmitters for sending notification signals. For example, WiMAX uses CDMA codes for its users to indicate presence and contend for bandwidth assignments [8]. SMACK [32] and FICA [74] also use orthogonal OFDM tones to send binary notification concurrently from multiple users. The use of random preamble sequences in MIMO/CON can be viewed as a special form of CDMA, but the codes do not need to be orthogonal, and their lengths can be further controlled by exploiting the sparsity with compressive sensing.

MIMO/CON's collision handling scheme is based on interference removal techniques, which has previously been exploited to address interference under various settings (e.g. [38, 55]). The novelty of MIMO/CON's approach lies in using concurrent channel estimation for packet identification. With compressive sensing decoding, MIMO/CON can reliably identify packet senders, while other approaches such as ZigZag [38] rely on the small frequency differences in oscillators between sender

hardware.

Finally, a recent proposal, Contrabass [83], also propose to allow concurrent access in MU-MIMO networks. However, Contrabass does not exploit the expected sparsity in concurrent channel estimation, and consequently suffers from a higher overhead in preamble length and decoding complexity.

4.8 Summary

In this chapter, we have proposed an ambitious scheme for achieving full utilization of uplink capacity offered by an AP equipped with many receive antennas. A key to our scheme is a novel channel estimation method in the PHY layer which can identify active senders and estimate CSI from concurrently received packet preambles. This is achieved under the assumption that senders are only loosely synchronized and not subject to mutual or central coordination. In the MAC layer, MIMO/CON maximizes channel utilization by exploiting normal MAC layer retransmission mechanism to recover otherwise undecodable packets in a collision. We believe the concurrent channel access and estimation schemes of this chapter, or similar approaches, are important for future high-throughput multiuser MIMO networks.

Chapter 5

A Centralized MAC Design Based on Compressive Sensing

In this chapter, we will discuss another use of sparse recovery that enables efficient centralized medium access control. A typical problem in centralized medium access control is the need of collecting the sending queue statistics on host stations for proper scheduling of transmissions. Given that the number of potential senders in the network can be large, but the number of senders that have nonempty output queues may be small at any moment, a polling approach that sequentially queries every station is inefficient. Similar to the use of sparse recovery to estimate channel statistics in Chapter 4, we will see that similar ideas are also applicable to designing a centralized MAC. Centralized medium access control has many advantages over the distributed schemes, such as better quality of service support, mitigating hidden terminal problems, and ensuring better short-term fairness. The proposed efficient strategy can fundamentally lead to a shift towards centralized MAC approaches for

wireless LANs.

5.1 Introduction

Compressive sensing is an emerging technology that has drawn considerable attention recently for its ability to acquire and extract critical information efficiently. It has found applications in various fields such as medical imaging, cognitive radio, wireless communication, and sensor networks (see, e.g., [33]). In particular, two features of compressive sensing are worth noting. First, generating compressive measurements is blind to the content of the signal to be compressed and has low computational complexity. Second, it is sufficient to capture the signal with a small number of compressive measurements, which is approximately proportional to its information content, i.e., its sparsity, not its length. Therefore, compressive sensing is attractive in large-scale distributed scenarios where coordination is substantial, and important information is sparse.

Wireless medium access control concerns scheduling of radio channels shared among a network of distributed hosts. The state-of-the-art 802.11 wireless LAN standard adopts a CSMA/CA random access method, whereby contention is resolved by a randomized backoff counter on every host station. The idle channel is won by the host whose backoff counter expires first. This method works well when there are relatively few hosts in the network; however it has limitations due to its distributed control paradigm. For example, collision avoidance relies on local carrier sensing. When carrier sensing cannot function properly, e.g., in the presence of hidden terminals, throughput will greatly degrade (see, e.g., [15]). Quality-of-service (QoS)

policies are also difficult to be implemented due to the lack of a central coordination.

While central coordination can potentially provide better scheduling, its efficient implementation however has been a challenge. For example, polling-based MAC protocols [71][2] may suffer from the high communication overhead, proportional to the number of hosts n in the network, or require complex scheduling to improve their efficiency. 802.11 PCF [2] is one example: it is rarely deployed in practice due to its overhead. The inefficiency of polling in part results from the fact that the central coordinator may poll a host that does not have data to send. It is wasteful because in a large network, only a few hosts may be contending for the channel at any given time. This suggests the use of a sparse vector to represent the hosts in the network where the contending hosts are the nonzero components. AP's polling operations can be viewed as constructing the sparse vector by checking its components one by one. In this paper, we propose CS-MAC, a compressive sensing based MAC protocol that allows a coordinator to check all components of the sparse vector at once by receiving only a small number of messages by the hosts.

CS-MAC can realize efficient centralized scheduling for three reasons. (1) Compressive sensing allows CS-MAC to identify contending hosts by receiving only a few compressive measurements with the number of measurements approximately proportional to the number of contending hosts. The communication overhead on the central coordinator thus can be minimized. (2) The compressive measurements of host channel access requests are formed in the air from concurrent transmissions. This analog approach eliminates the need of scheduling request transmissions, and thus enables fast measurement collection. (3) CS-MAC uses a distributed random access proto-

col to limit the number of contending hosts at a given time when many hosts have data to send. Consequently, the number of required compressive measurements does not need to be adaptive to the network contention level and can be set to a fixed constant. In addition, CS-MAC can scale with the number of hosts. The overhead of CS-MAC grows sub-linearly with n, assuming fast decoding algorithms [60] and parallel processing hardware [77] are used.

We summarize the main contributions of this paper as follows: (1) We use compressive sensing to implement a compressive requests/multiple grants MAC for wireless LANs. To our knowledge, we are the first in the literature to design a complete MAC solution based on compressive sensing. (2) We show an analog radio implementation incorporating a suite of low-overhead methods to address key issues such as analog compressive measurements formation and easy-to-implement synchronization. We demonstrate the practicality of host requests recovery on a hardware prototype. (3) We show through software simulation that CS-MAC can offer better performance in both throughput and fairness over two state-of-the-art protocols, 802.11 DCF and Idle Sense [42].

5.2 CS-MAC Design

In contrast to 802.11 DCF, CS-MAC takes a centralized approach to schedule radio channel access. In other words, a central coordinator is used to first learn distributed hosts' needs for channel access and then schedule transmission slots accordingly. For clarity, in this paper we assume a single wireless access point (AP) serving as the central coordinator, and all hosts in the wireless LAN are associated with the AP.

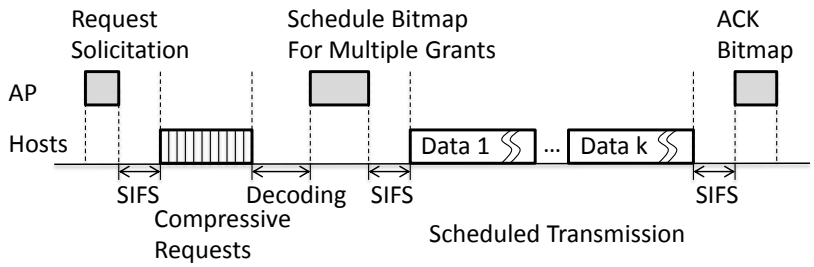


Figure 5.1: Overview of CS-MAC operations.

CS-MAC has two key ideas, namely the *compressive requests* and *multiple grants*. In CS-MAC, to acquire transmission opportunities, distributed hosts send channel access requests *concurrently*; thus multiple requests are combined in the air, which we call the compressive requests. The AP can then use the compressive requests to identify the contending hosts and grant transmission opportunities. As noted in Chapter 2, the number of measurements required for recovery is approximately proportional to the number of hosts requesting for channel access at the moment, rather than the total number of hosts in the network. CS-MAC thus can scale to networks with a large number of hosts.

Although the AP can efficiently learn the hosts' needs for channel access through compressive requests, given a fixed number of measurements, only a limited number of k hosts can request concurrently due to the sparsity constraint. To control the number of concurrent requesting hosts, CS-MAC uses a randomized scheme under which the hosts send requests with some probability. Noting that the AP can resolve up to k requesting hosts at a time and grant transmission opportunities to multiple hosts (called multiple grants). This contention thus is a *multi-winner* contention. We will show that in multi-winner contention, collisions are much less likely to occur

when k is sufficiently large, and thus the efficiency of CS-MAC is not hampered by the randomized scheme.

The basic operation of CS-MAC is outlined in Figure 5.1. CS-MAC begins with the AP initiating a request solicitation. Upon receiving the solicitation, hosts wish to access the channel may reply with requests, depending on the outcome of a local coin toss. The concurrent request transmissions then are combined in the air forming compressive requests. The AP next decodes the received compressive requests to identify contending hosts and schedules data transmissions accordingly. The schedule is then broadcast using a schedule bitmap packet. Finally, after the scheduled hosts finish transmitting data packets, the AP broadcasts an acknowledgement bitmap packet to acknowledge received packets.

5.2.1 Analog compressive requests: random linear combining in the air

As noted earlier, random projections of a sparse vector can preserve sufficient information with high probability for recovery. CS-MAC generates such random projections via concurrently transmitting random sequences as the requests from multiple hosts. The concurrent transmissions can be formulated as:

$$\mathbf{y} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \begin{bmatrix} h_1 x_1 \\ h_2 x_2 \\ \vdots \\ h_n x_n \end{bmatrix} + \mathbf{n} \tag{5.1}$$

where $\mathbf{a}_i$ is the random sequence of host i and h_i is the channel coefficient from host i to the AP. x_i is a binary {0,1} variable indicating whether host i sends its request and $\mathbf{n}$ denotes the noise vector. Assuming that $\mathbf{a}_i$ has length m and the channel is coherent over the transmission period, $\mathbf{y}$ is the received signal of length m at the AP.

For simplicity, the random sequence is generated from Bernoulli distribution of {1,-1}. Assuming there are at most k hosts requesting at any given time (we will justify this assumption later in Section 5.2.2), (5.1) can be viewed as a sparse recovery problem that has only k nonzero $h_i x_i$ in the unknown vector. Then m can be as small as $ck \ll n$ for exact recovery where c is a small constant. Empirically the value of c around 3 to 4 is sufficient for recovery provided that $\mathbf{n}$ is relatively small. Note that in identifying requesting hosts, the compressive sensing decoding process yields the solution of $h_i x_i$ without having to estimate the channel state information. Since $h_i x_i$ is 0 when host i does not request for channel access, we can use a threshold to distinguish between zero and nonzero x_i.

Finally, we note that the random sequence is assigned to each host by the AP during association. Therefore the AP knows the random sequence associated with each host and thus can solve (5.1) by using a proper sensing matrix in decoding. In addition, unlike many other centralized approaches, CS-MAC requires no sophisticated membership management at the AP. If a host leaves without notice, it is equivalent to the host not requesting for channel access. Since compressive sensing almost only concerns the number of nonzero components, a small increase in the total number of unknowns n resulting from loose membership management will almost not change the required number of measurements m.

5.2.2 Multi-winner contention for multiple grants

For exact recovery, the number of measurements m needs to be set based on the sparsity k, the number of requesting hosts. In the worst case where the network is very busy, k can be as high as n when all hosts need channel access. Therefore, without an adaptation scheme, a fixed large m would be required to support a potentially large k, resulting in inefficiency when in reality k is expected to be small. Furthermore, a larger m would not be practical if the longer measurement period exceeds the channel coherence time.

We propose to use a distributed control scheme to limit the sparsity k that CS-MAC needs to handle at any time, allowing CS-MAC to use a small fixed m. The basic idea is to use a random access protocol which stipulates the hosts to send requests with some probability p. p can be adjusted based on the network contention level: if collisions occur when more than k hosts send the requests, p will be reduced to avoid future collisions; otherwise p is increased for the hosts to take advantage of the unsaturated channel. The classic additive-increase-multiplicative-decrease (AIMD) principle can be employed to adjust p to ensure fairness among hosts. Note that the collisions can be detected by the AP when the decoding of compressive requests fails. The AP then can notify the hosts of the collision using the schedule bitmap so that they can adjust the requesting probability accordingly. We set the AIMD parameters similar to those in Idle Sense [42]: the probability is increased by 0.001 and decreased by $\frac{1}{1.2}$ for every adjustment.

It is possible that such a random access scheme suffers from excessive collisions and delivers low MAC efficiency. For example, slotted ALOHA only achieves at best

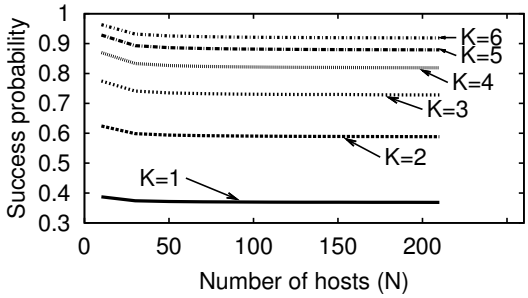


Figure 5.2: Advantages of multi-winner contention. The probability of successful resolution of contention increases dramatically when k grows from 1 to 5.

approximately 36% efficiency due to collisions. CS-MAC, however, does not suffer from the issue because multiple requesting hosts can be resolved for each compressive requests. To see this, we can derive the probability P_s that the requesting hosts are identified successfully as:

$$P_s = \sum_{i=1}^{K} \binom{n}{i} p^i (1-p)^{(n-i)} \tag{5.2}$$

Figure 5.2 shows the maximal P_s under different sparsity limit k and different number of hosts n. We can make the following observations. First, P_s remains almost constant under a varying number of hosts. Second, when $k = 1$ (corresponding to slotted ALOHA), the maximal P_s is only 0.36 as expected. However, when $k = 5$ the maximal P_s is dramatically increased to 0.9. This suggests that collisions are a lot less likely to occur when more than one winner is chosen in a contention. Also, a small number of winners is sufficient to mitigate the inefficiency significantly, and thus k needs not to be large. This can assure that the transmission time of compressive requests will be under channel coherence time. In our CS-MAC implementation, we set m=20 for k=5.

5.2.3 Synchronizing concurrent transmissions

To form compressive measurements from concurrent transmissions, all hosts need to be synchronized to the symbol-level. Fine-grained synchronization between hosts in such a distributed setting is generally a difficult task. CS-MAC instead only requires loose synchronization between hosts. It uses the request solicitation message as a reference signal. Assume a 300m radio range. Given this propagation delay, timing misalignments between transmitted symbols will be capped at $2\mu s$ [75].

To overcome the $2\mu s$ misalignment, we use a long symbol length of $5.12\mu s$ duration to ensure that the transmitted symbols always overlap, and the AP can safely take compressive measurements. Note that although the symbol is much longer than that in a perfectly synchronized scenario (e.g., in this case each symbol can be only 50ns long with a 20MHz bandwidth), the incurred overhead is still small due to the small number of symbols required for compressive requests. Given that CS-MAC only needs $m = 20$ symbols for compressive requests, the compressive requests span approximately $100\mu s$ duration. Since $k = 5$ hosts can be resolved and scheduled for data transmission after the compressive requests, CS-MAC on average only adds $20\mu s$ overhead to each data transmission, which corresponds to only two backoff slots in 802.11. Furthermore, the $100\mu s$ duration of the compressive requests is well below the 10-20ms channel coherence time, and thus (5.1) still holds.

5.2.4 Protocol overhead

Table 5.1 lists the parameter values of CS-MAC. Detailed descriptions of individual control packets are omitted due to space limitation. Compressive sensing decoding

may incur a significant overhead. For example, when linear programming is used to perform ℓ_1-norm minimization, the computational complexity is roughly $O(n^3)$ or higher. Reducing the decoding complexity has been a subject of intensive research in recent years [12]. Currently the best state-of-the-art algorithm can lower the decoding complexity down to $O(n \log \frac{n}{k})$ [13] at the expense of a relatively weak error guarantee for the recovered solution. Here we use $O(n^2)$ to approximate decoding complexity. This means the decoding time will be about $20\mu s$ with a 2GHz CPU when n=200.

Based on Table 5.1, and assuming the underlying physical layer runs 802.11g (54Mbps), the overhead for CS-MAC to send a data packet is $67.1\mu s$ excluding the overhead of random backoff and collisions in multi-winner contention. For 802.11 DCF, the overhead is $80.1\mu s$ without including the cost of random backoff and collisions. If the RTS-CTS mechanism is turned on, the overhead further goes up to $145.1\mu s$. We can see that while CS-MAC is a centralized approach for medium access control, its overhead is still comparable to basic 802.11 DCF, which only permits distributed random access.

Table 5.1: Parameter values of CS-MAC

Request solicitation	14 bytes	Decoding	20 μs
Compressive request	102.4 μs	Schedule bitmap	37 bytes
ACK bitmap	37 bytes	PHY header	20 μs

5.2.5 Performance gains of CS-MAC and system considerations

CS-MAC is a centralized MAC protocol, and thus has important gains over conventional CSMA-based protocols that are distributed in nature.

Hidden terminals. The classic hidden terminal problem arises when two hosts associated with the same AP cannot hear each other. As a result, they cannot detect ongoing transmissions and will interfere with each other. Throughput drops significantly when hidden terminals are present [15]. Current 802.11 protocol adopts the RTS-CTS exchange to avoid collisions, however its overhead is so high that RTS-CTS is often turned off. The hidden terminal problem arises because of a lack of global information at each host. Thus it is naturally solvable using a central scheduling approach such as CS-MAC. The scheduler guarantees a dedicated time slot for a host to send packets without interference. In an environment with multiple APs, there could be more complex hidden terminal scenarios where RTS-CTS cannot even function correctly [46], or exposed terminal scenarios which cause channel underutilization [15]. These problems can be solved by running CS-MAC on a central scheduler coordinating among the APs.

Short-term fairness. It is well-known that the binary exponential backoff scheme in 802.11 DCF delivers poor short-term fairness [42]. In general, to achieve good short-term fairness, one needs to estimate the network traffic load over small time intervals to prevent a single host from taking an unproportionally large portion of the channel. In a basic scenario where every host can hear each other, Idle Sense

[42] enforces the fairness by maintaining equal transmission probability at every host in a distributed fashion. The network load is estimated by observing the number of idle slots between transmissions. However, this estimation becomes difficult when hidden terminals exist. In this case, one may need to introduce additional control mechanisms to propagate load information [46]. CS-MAC takes a similar approach as Idle Sense that the hosts request for the channel with some probability. The network load then can be estimated by observing the number of requesting hosts, and the AP can simply use a broadcast to regulate host request probability. As a result, CS-MAC can achieve good short-term fairness regardless of hidden terminals in the sense of 802.11 DCF.

QoS. Centralized approaches can ease the implementation of quality of service (QoS) policies. For example, DOCSIS [5], the protocol for cable Internet access, is known for its ability to perform QoS scheduling for multimedia applications. We expect CS-MAC to deliver similar QoS capability since it takes a similar request/grant method. Take EDCF [4] in 802.11e as an example: EDCF provides differentiated service support by prioritizing flows in the network through adjustments to the contention window size for each flow. However, configuring priority-level parameters to achieve QoS is not obvious. In [82], proportional differentiation is proposed such that the ratio of channel sharing between different priority levels are to be set. To implement this policy, CS-MAC simply grants more transmission slots to higher priority flows.

Coexisting with 802.11. CS-MAC may coexist with 802.11. When conventional 802.11 DCF hosts join the network, they will not be granted any channel access under CS-MAC as they will not participate in CS-MAC requesting. We can solve

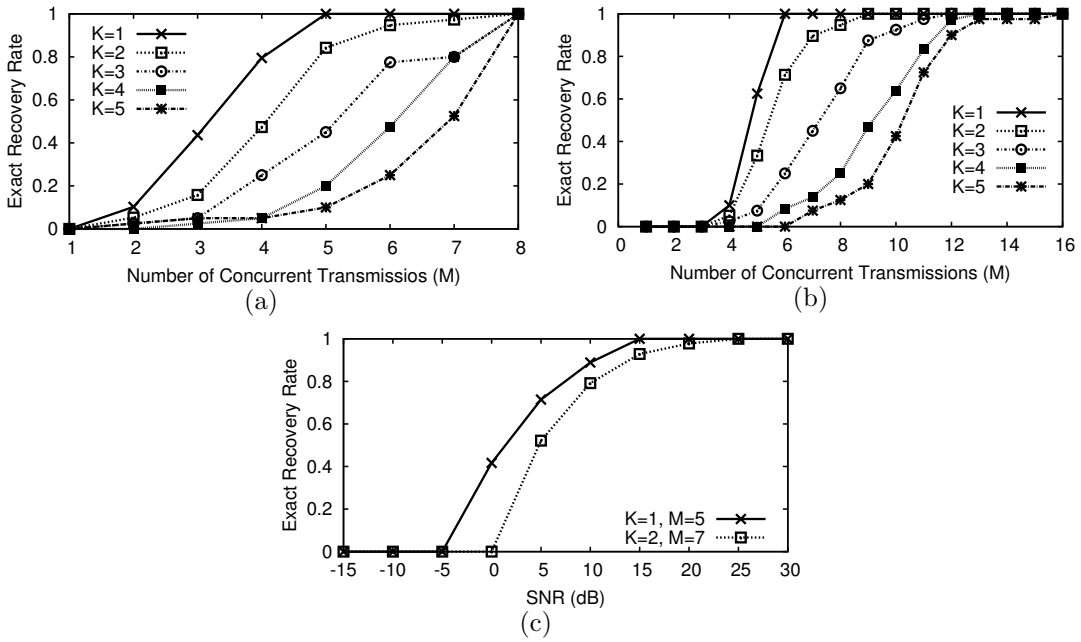


Figure 5.3: Experiment results on hardware prototype. (a) Recovery performance for compressive requests in the 8-node scenario, and that in (b) the 16-node scenario. (c) Recovery performance under different SNR in the 8-node scenario.

this problem by having the CS-MAC AP also perform CSMA before sending the request solicitation under CS-MAC. In other words, the AP contends with 802.11 hosts for the channel, and thus gives room for 802.11 hosts.

5.3 Compressive sensing recovery on hardware prototype

To demonstrate the practicality of analog compressive requests combined over air and their recovery by compressive sensing decoding, we implement these functions on software-defined radios. Our testbed has 9 USRP-N200 nodes distributed in an area of 2m×2m, equipped with the WBX daughterboards. The nodes operate with

a 0.78MHz bandwidth center at 916MHz. For fast prototyping, we calibrated the nodes before the experiments to ensure that there is no frequency offset. In practice, the AP's frequency can be used as a reference and the calibration can be done after receiving the request solicitation.

Among the 9 nodes, one node serves as the AP and the other 8 nodes serve as the hosts. Being near the AP, the hosts have a 20-30dB SNR. We randomly pick k hosts to request for channel access, and see if we can use compressive sensing decoding to recover the requests from the received signals. We use the `l1-MAGIC` package [3] to perform sparse recovery. The detection threshold for requests is set to 10dB to avoid false positive detection. Optimal threshold setting involves error and false positive/negative analysis, and needs further study.

Figure 5.3(a) shows the results of the experiment. When k=1, we can recover the requests exactly with m=5 concurrent transmissions or more. This is consistent with the $m \sim 4k$ estimate on the required measurements stated earlier. For $2 < k \leq 5$, we need 8 measurements for 100% recovery rate, but when fewer measurements are used, as m increases, we still can observe the increase in recovery rate. Next, we want to see how the recovery performs in a larger network setting. Due to limited hardware availability, we use the 8 USRP-N200 nodes to emulate a 16-node scenario. A single physical node will act as two different virtual nodes. The radio signals transmitted by the two virtual nodes are assumed to be perfectly combined without any noise. The virtually combined signal is then transmitted by the physical node. The results are shown in Figure 5.3(b). When k=2 and 3, requests are always recovered with 9 and 12 measurements, respectively.

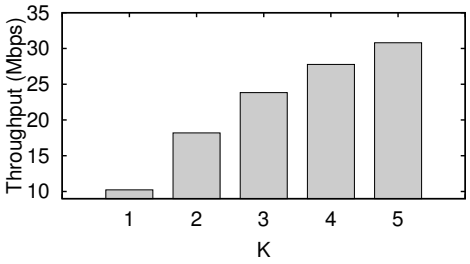


Figure 5.4: Impact of multi-winner contention. The aggregated throughput of CS-MAC in a 40-node scenario can reach 30Mbps when k increases from 1 to 5.

To check the performance under varying SNRs, we vary the transmission power in the 8-node scenario. Using k=1, we set m=5, the minimum measurements required for exact recovery as observed in Figure 5.3(a). Figure 5.3(c) shows the results. Compressive requests achieves good performance when SNR is higher than 15dB, a reasonable SNR requirement in wireless LANs. Similar performance is observed when k=2, m=7. Note that this parameter setting cannot guarantee 100% recovery as indicated by Figure 5.3(a), and thus shows a slightly worse performance.

5.4 Performance Simulation

To test a larger set of conditions, we implement CS-MAC with an event-driven software simulator. In the simulations, we assume the physical layer runs 802.11g (54Mbps), and all hosts always have data to send. We first show the impact of multi-winner contention in a 40-host scenario. Figure 5.4 shows the aggregated network throughput of CS-MAC with different k. The aggregated throughput of CS-MAC when k=1 is only 10Mbps due to the high collision probability. In contrast, the throughput is increased to 30Mbps by setting k=5 when collisions become less likely to occur. For all other simulations, k is set to 5 as described in Section 5.2.2.

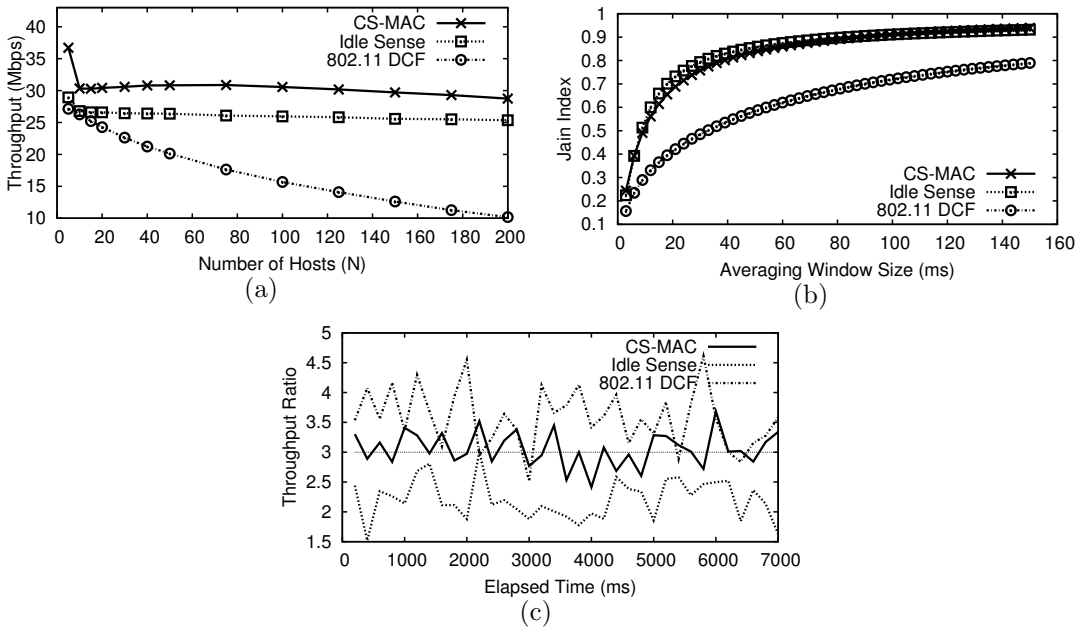


Figure 5.5: Software simulation results. (a) CS-MAC aggregated throughput with a varying number of hosts. (b) Short-term fairness [50]. (c) Proportional differentiated service for QoS.

Next we compare CS-MAC with Idle Sense [42] and 802.11 DCF in scalability to number of hosts, fairness, and QoS. Figure 5.5(a) shows the aggregated throughput of CS-MAC with a varying number of hosts. CS-MAC can scale to a network with 200 hosts without losing much of its efficiency, while 802.11 DCF loses its efficiency due to the increased number of collisions. Figure 5.5(b) shows the short-term fairness when n=40. CS-MAC achieves similar performance with Idle Sense. We note that when hidden terminals are introduced in the simulations, the throughput of both Idle Sense and 802.11 DCF drops significantly. If we enable RTS-CTS, they still suffer from poor short-term fairness due to the lack of correct network load estimation.

Lastly, Figure 5.5(c) shows the performance results of a simple implementation of a proportional differentiated service QoS policy. We want to have a particular host

in the 10-host collection to have throughput three times higher than the rest. For 802.11 DCF and Idle Sense, due to the lack of central coordination, we implement the policy by setting the contention window of the host to be three times smaller than other; however this implementation fails to achieve the correct ratio for both of the protocols. In contrast, CS-MAC can realize the ratio easily by AP scheduling.

5.5 Summary

CS-MAC of this chapter uses a central controller to schedule hosts. It is easy to see that a centralized approach can conveniently avoid hidden terminal problems, assure QoS and enhance fairness. However, it can be difficult to devise an efficient implementation for a centralized scheme due to the need of gathering global information from all hosts. In this paper, we note that compressive sensing can change the equation. Since host requests are expected to be sparse, they can now be recovered with far fewer measurements than before. This can fundamentally lead to a shift towards centralized MAC approaches for wireless LANs.

While in this chapter we have developed the basic concepts of CS-MAC and demonstrated its working under lab settings, we recognize that much further work is needed. In particular, a better understanding of the robustness of CS-MAC for networks with larger propagation delays and larger variation in host SNR is necessary. We also need to develop models for optimal choices of k and m for the contention period and for optimal design of the measurement matrix of (5.1) to accommodate more users.

Chapter 6

Conclusion

Sparse recovery is an exciting new research direction with many unexplored potential applications. Applications of sparse recovery can be broadly divided into two classes. First, the sparsity constraint is a powerful prior for data analysis and data processing. For example, imposing the sparsity constraint in feature learning encourages unsupervised discovery of localized, physically meaningful data patterns. In addition, sparse recovery methods can be employed for constructing sparse data representations, which are more linearly separable in the feature space and can be useful for machine learning tasks. Second, given sparse data, one can construct low-dimensional and compressed data representations through simple linear projections, and expect that data reconstruction will be successful using sparse recovery methods. The projection operations can be further incorporated into the data acquisition process, in which the amount of sampling operations and the number of samples acquired can be dramatically reduced.

The power of sparse recovery, however, can be hampered by the high computation

cost associated with the recovery algorithms. In some application scenarios, one may have a very short time budget for obtaining a solution, or the amount of data to process is simply too big to afford a computationally expensive solver. Convex optimization based approaches, known for their high computation costs, are impractical for these applications. On the other hand, although greedy iterative methods can find solutions very efficiently, these solutions are often unsatisfactory due to their weak stability under noise.

In this thesis, we have found that the greedy approaches can be brought back to the table by leveraging application insights. In particular, the stability of greedy sparse recovery algorithms can be largely improved by adding new application-specific constraints, even very simple ones. In addition, given that the greedy algorithms usually contain only a few simple steps, incorporating new constraints into the algorithms is often not difficult. We provide two example applications in this thesis, and demonstrate that this approach indeed can improve the performance of greedy recovery algorithms. More specifically, some of the main findings in this thesis are:

1. In computing sparse feature-space representations for image data, imposing the nonnegativity constraint to the feature dictionary and representations allows OMP to find very stable representations. As a result, image classification using these representations delivers high accuracy, outperforming those using the classical unconstrained OMP encoder by large margins.

2. In identifying participating senders from overlapping wireless symbol sequences, the receive antenna diversity on a base station can be leveraged to provide a "same-support" constraint to the recovery algorithms. By incorporating this

constraint, the greedy algorithms can be made to converge in very few iterations.

Overall, the take-home message of this work is the importance of leveraging new constraints when designing greedy sparse recovery algorithms and developing new applications. Simple constraints often can make a difference in recovery stability, which may directly impact the performance of a given application. With this approach, computationally efficient methods such as OMP can be made very competitive.

Bibliography

[1] Ettus research, [online] 2012, www.ettus.com.

[2] IEEE standard 802.11-2007.

[3] l1-MAGIC. Available at www.l1-magic.org.

[4] IEEE 802.11e/D5.0. 2003.

[5] Cable television laboratories, inc. data-over-cable service interface specifications, in radio frequency interface specification. 2004.

[6] IEEE 802.11n-2009. October 2009.

[7] Jorgen Bach Andersen, Theodore S. Rappaport, and Susumu Yoshida. Propagation measurements and models for wireless communications channels. *IEEE Communications Magazine*, 33(1):42–49, 1995.

[8] Jeffrey G. Andrews, Arunabha Ghosh, and Rias Muhamed. *Fundamentals of WiMAX: understanding broadband wireless networking*. Prentice Hall, 2007.

[9] Waheed U. Bajwa, Jarvis Haupt, Akbar M. Sayeed, and Robert Nowak. Compressed channel sensing: A new approach to estimating sparse multipath channels. *Proceedings of the IEEE*, 98(6):1058–1076, 2010.

[10] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[11] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, August 2013.

[12] Radu Berinde, Anna C Gilbert, Piotr Indyk, Howard Karloff, and Martin J Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Proceedings of Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2008.

[13] Radu Berinde, Piotr Indyk, and Milan Ruzic. Practical near-optimal sparse recovery in the l1 norm. In *Proceedings of Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2008.

[14] Pietro Berkes and Laurenz Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5(6):579–602, 2005.

[15] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: a media access protocol for wireless LANs. In *Proceedings of ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*. ACM, 1994.

[16] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2011.

[17] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for RGB-D based object recognition. In *Proceedings of International Symposium on Experimental Robotics (ISER)*, 2012.

[18] Alfred M. Bruckstein, David L. Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.

[19] Alfred M. Bruckstein, Michael Elad, and Michael Zibulevsky. On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Transactions on Information Theory*, 54(11):4813–4820, 2008.

[20] Emmanuel J. Candes and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[21] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[22] Adam Coates and Andrew Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of International Conference on Machine Learning (ICML)*, 2011.

[23] Adam Coates and Andrew Y. Ng. Selecting receptive fields in deep networks. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2011.

[24] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[25] Mark A. Davenport, Petros T. Boufounos, Michael B. Wakin, and Richard G. Baraniuk. Signal processing with compressive measurements. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):445–460, 2010.

[26] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[27] David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[28] David L. Donoho, Michael Elad, and Vladimir N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.

[29] David L. Donoho and Xiaoming Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862, 2001.

[30] David L. Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2003.

[31] David L. Donoho, Yaakov Tsaig, Iddo Drori, and J-L. Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 58(2):1094–1121, 2012.

[32] Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker. SMACK: a smart acknowledgment scheme for broadcast messages in wireless networks. In *Proceedings of ACM Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2009.

[33] Yonina C. Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[34] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[35] Robert Gens and Pedro Domingos. Discriminative learning of sum-product networks. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2012.

[36] David Gesbert, Marios Kountouris, Robert W. Heath, Chan-Byoung Chae, and Thomas Salzer. Shifting the MIMO paradigm. *IEEE Signal Processing Magazine*, 24(5):36–46, 2007.

[37] Saeed S. Ghassemzadeh, Rittwik Jana, Christopher W. Rice, William Turin, and Vahid Tarokh. Measurement and modeling of an ultra-wide bandwidth indoor channel. *IEEE Transactions on Communications*, 52(10):1786–1796, 2004.

[38] Shyamnath Gollakota and Dina Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *Proceedings of ACM Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2008.

[39] Ian Goodfellow, Aaron Courville, and Yoshua Bengio. Large-scale feature learning with spike-and-slab sparse coding. In *Proceedings of International Conference on Machine Learning (ICML)*, 2012.

[40] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout Networks. In *Proceedings of International Conference on Machine Learning (ICML)*, 2013.

[41] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

[42] Martin Heusse, Franck Rousseau, Romaric Guillier, and Andrzej Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs. In *Proceedings of ACM Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2005.

[43] Patrik O. Hoyer. Modeling receptive fields with non-negative sparse coding. *Neurocomputing*, 52:547–552, 2003.

[44] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[45] Ka-yu Hui. Direct modeling of complex invariances for visual object features. In *Proceedings of International Conference on Machine Learning (ICML)*, 2013.

[46] Ying Jian and Shigang Chen. Can CSMA/CA networks be made fair? In *Proceedings of ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2008.

[47] William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(1):189–206, 1984.

[48] Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. Fast inference in sparse coding algorithms with applications to object recognition. *CBLL-TR-2008-12-01, New York University*, 2008.

[49] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann LeCun. Learning convolutional feature hierarchies for visual recognition. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2010.

[50] Can Emre Koksal, Hisham Kassab, and Hari Balakrishnan. An analysis of short-term fairness in wireless media access protocols. In *Proceedings of ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2000.

[51] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

[52] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[53] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2006.

[54] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of International Conference on Machine Learning (ICML)*, 2009.

[55] Li Erran Li, Kun Tan, Harish Viswanathan, Ying Xu, and Yang Richard Yang. Retransmission≠ repeat: simple retransmission permutation can resolve overlapping channel collisions. In *Proceedings of ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2009.

[56] Kate Ching-Ju Lin, Shyamnath Gollakota, and Dina Katabi. Random access heterogeneous MIMO networks. In *Proceedings of ACM Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2011.

[57] Eugenio Magistretti, Krishna Kant Chintalapudi, Bozidar Radunovic, and Ramachandran Ramjee. WiFi-Nano: reclaiming WiFi efficiency through 800 ns slots. In *Proceedings of ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2009.

[58] Thomas L. Marzetta. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Transactions on Wireless Communications*, 9(11):3590–3600, 2010.

[59] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

[60] Deanna Needell and Joel A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[61] Deanna Needell and Roman Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of Computational Mathematics*, 9(3):317–334, 2009.

[62] Jiquan Ngiam, Pang Wei Koh, Zhenghao Chen, Sonia A. Bhaskar, and Andrew Y. Ng. Sparse filtering. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2011.

[63] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[64] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[65] Yagyensh Chandra Pati, Ramin Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of Asilomar Conference on Signals, Systems and Computers*. IEEE, 1993.

[66] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 759–766, 2007.

[67] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann Lecun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007.

[68] Justin Romberg. An overview of recent results on the identification of sparse channels using random probes. In *Proceedings of IEEE Conference on Decision and Control (CDC)*. IEEE, 2010.

[69] Christopher J. Rozell, Don H. Johnson, Richard G. Baraniuk, and Bruno A. Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563, 2008.

[70] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 2008.

[71] Oran Sharon and Eitan Altman. An efficient polling MAC for wireless LANs. *IEEE/ACM Transactions on Networking*, 9(4):439–451, 2001.

[72] Vikas Sindhwani and Amol Ghoting. Large-scale distributed non-negative sparse coding and sparse dictionary learning. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2012.

[73] Martin Slawski and Matthias Hein. Sparse recovery by thresholded non-negative least squares. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2011.

[74] Kun Tan, Ji Fang, Yuanyang Zhang, Shouyuan Chen, Lixin Shi, Jiansong Zhang, and Yongguang Zhang. Fine-grained channel access in wireless LAN. In *Proceedings of ACM Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2010.

[75] Kun Tan, Ji Fang, Yuanyang Zhang, Shouyuan Chen, Lixin Shi, Jiansong Zhang, and Yongguang Zhang. Fine-grained channel access in wireless LAN. In *Proceedings of ACM Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2010.

[76] Kun Tan, He Liu, Ji Fang, Wei Wang, Jiansong Zhang, Mi Chen, and Geoffrey M. Voelker. SAM: enabling practical spatial multiple access in wireless LAN. In *Proceedings of ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2009.

[77] Stephen J. Tarsa, Tsung-Han Lin, and H. T. Kung. Performance gains in conjugate gradient computation with linearly connected gpu multiprocessors. In *Proceedings of USENIX Workshop on Hot Topics in Parallelism (HotPar)*, 2012.

[78] Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[79] Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[80] Davd Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge University Press, 2005.

[81] J-J. Van de Beek, Ove Edfors, Magnus Sandell, Sarah Kate Wilson, and P. Ola Borjesson. On channel estimation in OFDM systems. In *Proceedings of IEEE Vehicular Technology Conference (VTC)*, 1995.

[82] Qi Xue and Aura Ganz. Proportional service differentiation in wireless LANs using spacing-based channel occupancy regulation. In *Proceedings of ACM international conference on Multimedia (MM)*. ACM, 2004.

[83] Sungro Yoon, Injong Rhee, Bang Chul Jung, Babak Daneshrad, and Jae H. Kim. Contrabass: Concurrent transmissions without coordination for ad hoc networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 1134–1142. IEEE, 2011.

[84] Matthew D. Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *Proceedings of International Conference on Learning Representations (ICLR)*, 2013.